12. Abrahamsson, P., Warsta, J., Siponen, M.T., & Ronkainen, J. (2003). New Directions on Agile Methods: A Comparative Analysis. *Proceedings of ICSE'03*, 244-254
13. Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile Software Development Methods: Review and Analysis. *VTT Publications 478*
14. Aydin, M.N., Harmsen, F., Slooten van K., & Stegwee, R.A. (2005). On the Adaptation of An Agile Information(Suren) Systems Development Method. *Journal of Database Management Special issue on Agile Analysis, Design, and Implementation, 16(4),* 20-24
15. "David Bock's Weblog : Weblog". Jroller.com. http://jroller.com/page/bokmann?entry=improving_your_processes_aim_high. Retrieved 2010-04-02.
16. "Agility measurement index". Doi.acm.org. http://doi.acm.org/10.1145/1185448.1185509. Retrieved 2010-04-02.
17. Peter Lappo; Henry C.T. Andrew. "Assessing Agility". http://www.smr.co.uk/presentations/measure.pdf. Retrieved 2010-06-06.
18. Kurian, Tisni (2006). "Agility Metrics: A Quantitative Fuzzy Based Approach for Measuring Agility of a Software Process" *ISAM-Proceedings of International Conference on Agile Manufacturing'06(ICAM-2006)*, Norfolk, U.S.
19. Joe Little (2007-12-02). "Nokia test, A Scrum specific test". Agileconsortium.blogspot.com. http://agileconsortium.blogspot.com/2007/12/nokia-test.html. Retrieved 2010-06-06.
20. Mark Seuffert, Piratson Technologies, Sweden. "Karlskrona test, A generic agile adoption test". Piratson.se. http://www.piratson.se/archive/Agile_Karlskrona_Test.html. Retrieved 2010-06-06.
21. "How agile are you, A Scrum specific test". Agile-software-development.com. http://www.agile-software-development.com/2008/01/how-agile-are-you-take-this-42-point.html. Retrieved 2010-06-06.
22. "Agile Methodologies Survey Results" (PDF). Shine Technologies. 2003. http://www.shinetech.com/attachments/104_ShineTechAgileSurvey2003-01-17.pdf. Retrieved 2010-06-03. "95% [stated] that there was either no effect or a cost reduction . . . 93% stated that productivity was better or significantly better . . . 88% stated that quality was better or significantly better . . . 83% stated that business satisfaction was better or significantly better"
23. Ambler, Scott (August 3, 2006). "Survey Says: Agile Works in Practice". *Dr. Dobb's*. http://www.drdobbs.com/architecture-and-design/191800169;jsessionid=2QJ23QRYM3H4PQE1GHPCKH4ATMY32JVN?queryText=agile+survey. Retrieved 2010-06-03. "Only 6 percent indicated that their productivity was lowered . . . No change in productivity was reported by 34 percent of respondents and 60 percent reported increased productivity. . . . 66 percent [responded] that the quality is higher. . . . 58 percent of organizations report improved satisfaction, whereas only 3 percent report reduced satisfaction."
24. "The State of Agile Development" (PDF). VersionOne, Inc.. 2008. http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf. Retrieved 2010-07-03. "Agile delivers"
25. "Answering the "Where is the Proof That Agile Methods Work" Question". Agilemodeling.com. 2007-01-19. http://www.agilemodeling.com/essays/proof.htm. Retrieved 2010-04-02.
26. Agile Processes Workshop II Managing Multiple Concurrent Agile Projects. Washington: OOPSLA 2002
27. W. Scott Ambler (2006) "Supersize Me (http://www.drdobbs.com/184415491)" in Dr. Dobb's Journal, February 15, 2006.
28. Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Boston, MA: Addison-Wesley. ISBN 0-321-27865-8.
29. Boehm, B.; R. Turner (2004). *Balancing Agility and Discipline: A Guide for the Perplexed*. Boston, MA: Addison-Wesley. pp. 55–57. ISBN 0-321-18612-5.
30. Schaaf, R.J. (2007). "Agility XL", Systems and Software Technology Conference 2007 (http://www.sstc-online.org/Proceedings/2007/pdfs/RJS1722.pdf), Tampa, FL
31. "Bridging the Distance". Sdmagazine.com. http://www.drdobbs.com/architecture-and-design/184414899. Retrieved 2011-02-01.
32. Martin Fowler. "Using an Agile Software Process with Offshore Development". Martinfowler.com.

http://www.martinfowler.com/articles/agileOffshore.html. Retrieved 2010-06-06.
33. [The Art of Agile Development James Shore & Shane Warden pg 47]
34. [1] (http://www.logigear.com/in-the-news/973-agile.html) LogiGear, PC World Viet Nam, Jan 2011
35. 2000 (http://ciclamino.dibe.unige.it/xp2000/)
36. "2006". Virtual.vtt.fi. http://virtual.vtt.fi/virtual/xp2006/. Retrieved 2010-06-06.
37. "2010". Xp2010.org. http://www.xp2010.org/. Retrieved 2010-06-06.
38. 2001 (http://www.xpuniverse.com/2001/xpuPapers.htm)
39. 2002 (http://www.xpuniverse.com/2002/schedule/schedule)
40. 2003 (http://www.xpuniverse.com/2003/schedule/index)
41. 2004 (http://www.xpuniverse.com/2004/schedule/index)
42. "Agile Development Conference". Agile200x.org. http://www.agile200x.org/. Retrieved 2010-06-06.

Further reading

- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile Software Development Methods: Review and Analysis. *VTT Publications 478*.
- Cohen, D., Lindvall, M., & Costa, P. (2004). An introduction to agile methods. In *Advances in Computers* (pp. 1–66). New York: Elsevier Science.
- Dingsøyr, Torgeir, Dybå, Tore and Moe, Nils Brede (ed.): *Agile Software Develoment: Current Research and Future Directions* (http://www.amazon.co.uk/Agile-Software-Development-Research-Directions/dp/3642125743), Springer, Berlin Heidelberg, 2010.
- Fowler, Martin. *Is Design Dead?* (http://www.martinfowler.com/articles/designDead.html). Appeared in *Extreme Programming Explained*, G. Succi and M. Marchesi, ed., Addison-Wesley, Boston. 2001.
- Larman, Craig and Basili, Victor R. *Iterative and Incremental Development: A Brief History* IEEE Computer, June 2003 (http://www.highproductivity.org/r6047.pdf)
- Riehle, Dirk. *A Comparison of the Value Systems of Adaptive Software Development and Extreme Programming: How Methodologies May Learn From Each Other* (http://www.riehle.org/computer-science/research/2000/xp-2000.html). Appeared in *Extreme Programming Explained*, G. Succi and M. Marchesi, ed., Addison-Wesley, Boston. 2001.
- Rother, Mike (2009). *Toyota Kata*. McGraw-Hill. ISBN 0071635238. http://books.google.com/?id=_1lhPgAACAAJ&dq=toyota+kata
- M. Stephens, D. Rosenberg. *Extreme Programming Refactored: The Case Against XP*. Apress L.P., Berkeley, California. 2003. ISBN 1-59059-096-1

External links

- Manifesto for Agile Software Development (http://www.agileManifesto.org/)
- The Agile Alliance (http://www.agilealliance.org/)
- The Agile Executive (http://theagileexecutive.com/)
- Article Two Ways to Build a Pyramid by John Mayo-Smith (http://www.informationweek.com/news/software/development/showArticle.jhtml?articleID=6507351)
- Agile Software Development: A gentle introduction (http://www.agile-process.org/)
- The New Methodology (http://martinfowler.com/articles/newMethodology.html) Martin Fowler's description of the background to agile methods
- Agile Journal (http://www.agilejournal.com/) - Largest online community focused specifically on agile development
- [9] (http://www.dmoz.org%7CComputers/Programming/Methodologies/Agile%7CAgile)]
- Agile Cookbook (http://agilecookbook.com/)
- Ten Authors of The Agile Manifesto Celebrate its Tenth Anniversary (http://www.pragprog.com/magazines/2011-02/agile--)

Standards

There are a few industry standards related to process improvement models we should mention briefly. For you as a beginner, it is enough to know they exist. However, if you start working for large corporations, you will find that many will follow one or the other of these standards.

Capability Maturity Model Integration

The Capability Maturity Model Integration (CMMI) is one of the leading models and based on best practice. Independent assessments grade organizations on how well they follow their defined processes, not on the quality of those processes or the software produced. CMMI has replaced CMM.

ISO 9000

ISO 9000 describes standards for a formally organized process to manufacture a product and the methods of managing and monitoring progress. Although the standard was originally created for the manufacturing sector, ISO 9000 standards have been applied to software development as well. Like CMMI, certification with ISO 9000 does not guarantee the quality of the end result, only that formalized business processes have been followed.

ISO 15504

ISO 15504, also known as Software Process Improvement Capability Determination (SPICE), is a "framework for the assessment of software processes". This standard is aimed at setting out a clear model for process comparison. SPICE is used much like CMMI. It models processes to manage, control, guide and monitor software development. This model is then used to measure what a development organization or project team actually does during software development. This information is analyzed to identify weaknesses and drive improvement. It also identifies strengths that can be continued or integrated into common practice for that organization or team.

External Links

- CMMI Official Website (http://www.sei.cmu.edu/cmmi)
- Capability Maturity Model (http://www.dmoz.org/Computers/Programming/Methodologies/Capability_Maturity_Model/) at DMOZ
- ISO 9000 (http://www.dmoz.org/Science/Reference/Standards/Individual_Standards/ISO/ISO_9000/) at DMOZ
- Introduction to ISO 9000 and ISO 14000 (http://www.iso.org/iso/iso_catalogue/management_standards/iso_9000_iso_14000.htm)
- ISO 15504 News (isospice) (http://www.isospice.com)
- Automotive SPICE (http://www.automotivespice.com/)

Life Cycle

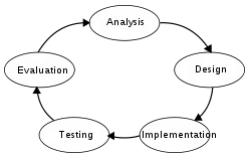
The **Systems Development Life Cycle (SDLC)**, or *Software Development Life Cycle* in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. The concept generally refers to computer or information systems.

Model of the Systems Development Life Cycle

In software engineering the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system[1]: the software development process.

Overview

Systems Development Life Cycle (SDLC) is a process used by a systems analyst to develop an information system, including requirements, validation, training, and user (stakeholder) ownership. Any SDLC should result in a high quality system that meets or exceeds customer expectations, reaches completion within time and cost estimates, works effectively and efficiently in the current and planned Information Technology infrastructure, and is inexpensive to maintain and cost-effective to enhance.[2]



Model of the Systems Development Life Cycle

Computer systems are complex and often (especially with the recent rise of Service-Oriented Architecture) link multiple traditional systems potentially supplied by different software vendors. To manage this level of complexity, a number of SDLC models have been created: "waterfall"; "fountain"; "spiral"; "build and fix"; "rapid prototyping"; "incremental"; and "synchronize and stabilize". [3]

SDLC models can be described along a spectrum of agile to iterative to sequential. Agile methodologies, such as XP and Scrum, focus on light-weight processes which allow for rapid changes along the development cycle. Iterative methodologies, such as Rational Unified Process and Dynamic Systems Development Method, focus on limited project scopes and expanding or improving products by multiple iterations. Sequential or big-design-up-front (BDUF) models, such as Waterfall, focus on complete and correct planning to guide large projects and risks to successful and predictable results[*citation needed*]. Other models, such as Anamorphic Development, tend to focus on a form of development that is guided by project scope and adaptive iterations of feature development.