Project Management Life Cycle Approaches For Software Development

Marjan Angeleski[1] Olivera Kostoska[2] Dejan Zdraveski[3]

[1]Marjan Angeleski, Faculty of economics, Prilep, University "St. Kliment Ohridski"-Bitola, Republic of North Macedonia, marjan.angeleski@uklo.edu.mk, https://orcid.org/0000-0002-4250-3517

[2]Olivera Kostoska, Faculty of economics, Prilep, University "St. Kliment Ohridski"-Bitola, Republic of North Macedonia, olivera.kostoska@uklo.edu.mk, https://orcid.org/0000-0002-4794-635X

[3]Dejan Zdraveski, Faculty of economics, Prilep, University "St. Kliment Ohridski"-Bitola, Republic of North Macedonia, dejan.zdraveski@uklo.edu.mk, https://orcid.org/0009-0005-2999-4330

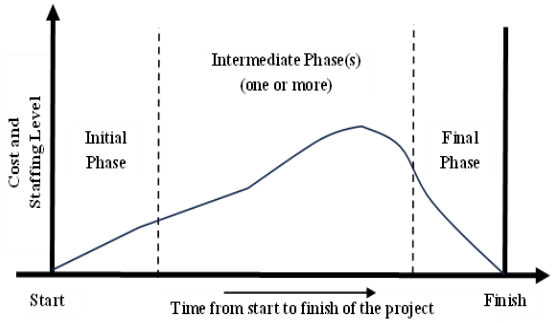
Abstract: The project from its beginning to its finalization goes through different phases that together form the so-called project life cycle. The paper analyzes four basic project management life cycle approaches that are applied especially in the management of software projects: predictive project management life cycle approach; Iterative project management model; Incremental project management approach and Agile project management approach. The literature review shows that, despite the fact that agile methodologies such as Scrum are the most preferred in software development, in certain cases, especially for projects with stable, low risk environments, etc., other approaches can be used effectively and efficiently.

Keywords: software development, project management, life cycle, life cycle approaches

1. PROJECT LIFE CYCLE AND TYPES OF PROJECT MANAGEMENT LIFE CYCLE

There are many different definitions of a project, but basically a project is "a set of new, conjugated, specific, methodically structured activities whose aim is to achieve an objective stated in a defined period and that implies development of innovative activities." (Aurelia, 2014). A project from its start to its completion goes through different phases that together form the so-called project life cycle. A project life cycle is made up of phases that connect the delivery of business and stakeholder value from the beginning to the end of the project. (PMI, 2021). Different authors note different titles of the phases within the project life cycle, but the following phases are generally mentioned: 'Starting', 'Preparing', 'Carrying Out' and 'Ending' (Wang & Chen, 2023). According to the first version of the PMBOK, the typical or generic project life cycle begins with an initial phase, continues with intermediate phase(s) that may include one or more phases, and ends with a finale phase (Figure 1) (PMI, 1996).

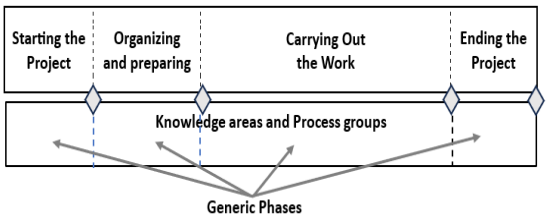
Figure 1. Generic Life Cycle



Source: Author's adaptation according PMI. (1996). A Guide to the Project Management Body of Knowledge. Project Management Institute

In later versions of the PMBOK, taking into account that projects can vary in size and the amount of complexity, the project follows the following phases: "Starting the project", "Organizing and preparing", "Carrying out the work", and "Closing the project" (Figure 2) (PMI, 2017).

Figure 2. Generic Life Cycle



Source: Author's adaptation according PMI. (2017). A Guide to the Project Management Body of Knowledge. Project Management Institute

According to Kerzner, there is no unified approach to defining the life-cycle phases of a project, not only between different industries but also within the same industry (Kerzner, 2009). Labuschagne, systematizing the different project life-cycles proposed by various researchers, defines seven possible generic life cycle phases in a project: Idea generation, Pre-feasibility, Feasibility, Development and execution, Commissioning, Launch, Post Implementation Review (PIR) (Labuschagne, 2005), while according to Watt at al., a standard project typically has the following four major phases: initiation, planning, implementation, and closure (Watt et al., 2014).

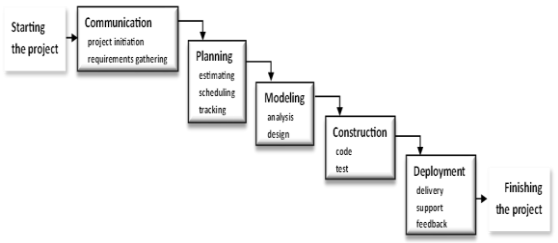
Specifically, when it comes to the life-cycle phases for computer programming, Kerzner defines the following phases: conceptual phase; planning phase; definition and design phase; implementation phase; conversion phase, while for construction companies, he generally lists the following phases: Planning, data gathering, and procedures; Studies and basic engineering; Major review; Detail engineering; Detail engineering/construction overlap; Construction; Testing and commissioning, despite stating that even in this sector that is already established in the realization of projects, ten different construction companies could have different definitions for the life-cycle phases (Kerzner, 2009).

Basically, project life cycles can be predictive or adaptive with different phases that participate in the development of the product, service or result. Such Development life cycles can be: predictive, iterative, incremental, adaptive and hybrid as a combination of some of the previous ones. (PMI, 2017)

2. PREDICTIVE PROJECT MANAGEMENT LIFE CYCLE APPROACHE (WATERFALL)

Waterfall model is a typical representative of the predictive project management life cycle approach and represents a sequential approach to project management that follows a step-by-step linear and predefined sequence of activities. It got its name "waterfall" because its visual representation resembles a waterfall cascade. In this approach, the end of the previous phase of the project plan is an input to the beginning of the next phase and "once a phase is completed, it can be difficult and costly to revisit a previous stage." (Radigan, 2024). This means that after the completion of the phase, it is usually not subject to revision because changes require high implementation costs and the iteration itself would mean re-implementation of the activities (Petersen et al., 2009). Therefore, deliverables from the previous phase are usually approved before the next phase begins, although in certain cases, if it is an acceptable risk, the next phase can begin before it is confirmed that the deliverables from the previous phase have been accepted (PMI, 1996). The dependencies between the defined activities and phases are known already in the planning process and they are monitored throughout the entire project life cycle. This approach focuses on the plan that should be carefully conceived in order to achieve the desired results through its effective and efficient execution. This type of project management belongs to predictive project management, because, as previously noted, it is an approach with clearly defined expected results and clearly planned and predictable activities in advance. Since the requirements and final results are fixed, most of the management falls on the time management and cost management of the project. Although this approach was initially proposed in the development of software projects (Bell & Thayer, 1976) with five main phases (requirements analysis and specification, design, implementation, and unit testing, integration and system testing, and operation and maintenance) (Abdul et al., 2024), the modern project management methodologies are based on the Waterfall approach which is "…relevant and applicable to all projects where the project scope and the time and cost associated…" (Van Der Merwe, 2017). The Waterfall approach is most suitable for projects that do not require iterative or incrementality in the implementation of project activities and "…this is suitable for stable, low risk environments." (APM, 2019), i.e. for projects that have a clear representation of the final product (Adobe, 2021). In the construction industry, where the components of the project are clearly defined in detailed documentation and where it is highly likely that the requirements will not change during the implementation of the activities, the waterfall methodology is particularly effective (Balasubramaniam et al., 2023). Pressman, who describes the waterfall model in software development, defines that waterfall work-flow "…begins with customer specification of requirements and progresses through planning, modeling, construction, and deployment, culminating in ongoing support…" (Pressman, 2010). In this model, in the first phase, "Communication", which is a very important initial phase of the project, it describes that it is necessary to gather requirements from customers and initiate the project processes; in the "Planning" phase, estimating project resources and time is performed, as well as scheduling and tracking the schedule of activities; the next phase, "Modeling", includes analysis and design through a graphic or technical representation of the processes; in the next phase, "Construction", the development of the product begins, i.e. coding and testing of the code is performed to examine the functionality of the product; the project life cycle phases end with "Deployment" when a functional product ready for use is delivered, support for its use and feedback is provided in order to improve the product (Figure 3). Considering that the completion of the previous phase is an input to the next, there is strict control, approval by the project management and acceptance by the users after the completion of each phase (Lalband & Kavitha, 2019).

Figure 3. The waterfall model for software development



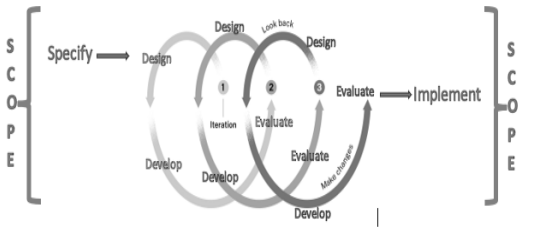
Source: Author's adaptation according Pressman, R. S. (2010). The Waterfall Model. In Software Engineering: A Practitioner's Approach (Seventh Ed). McGraw-Hill

3. ITERATIVE PROJECT MANAGEMENT MODEL

The iterative process is the practice of building, refining, and improving a project, product, or initiative until the project team and customers are satisfied with the end result (Asana, 2025). When following an iterative approach, the project scope is usually defined early in the project life cycle, while the time and cost estimates change over time as the project team's understanding of the product increases. Iterativity means that the project is divided into cycles or "iterations" and the final product is developed through multiple iterations. Each iteration aims to develop a partially functional version of the product, while the final product created by the project has functionality that is in line with the requirements and needs of stakeholders (PMI, 2017; Twproject, 2025). With each new iteration, the project creates improved functionality of the product or service being developed by the project, as well as a more advanced version of the product and/or service. This model is applicable to projects that may undergo changes during their life cycle, and it involves continuous communication and collaboration between the project manager, the project team, and external project stakeholders.

The generic Iterative project management life cycle model is shown in Figure 4 and includes five phases within the scope of the project: Specifying the requirements; Designing the product to meet those requirements; Developing the product; Evaluate the delivered system/product; Implementing the system by delivering to the end-users (Jenkins, 2003).

Figure 4. Phases of the iterative project lifecycle (author's adaptation according to Jenkins, N. (2003). The Iterative Project Model (IPM) - A lightweight methodology for project management.

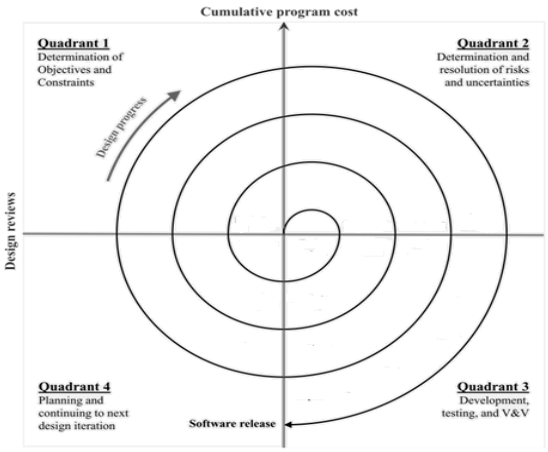


https://www.stickyminds.com/ & https://digital.gov/guides/hcd/design-concepts/iteration/)

As a representative of the iterative model, we can mention the spiral model, which is most widely used in software development and was initially developed by Boehm in 1986 with the aim of improving the software process model situation and creating a "risk-driven approach to the software process rather than a primarily document-driven or code-driven process" (Boehm, 1986).

The spiral model is graphically represented in Figure 5, shown in the Cartesian coordinate system in which the y-axis represents design review and the x-axis represents cumulative program cost, defining four quadrants "through which the lifecycle design process must progress,": (Quadrant 1) Determination of Objectives and Constraints; (Quadrant 2) Determination and resolution of risks and uncertainties; (Quadrant 3) Development, testing, and V&V; (Quadrant 4) Planning and continuing to next design iteration (Gujarathi et al., 2024).

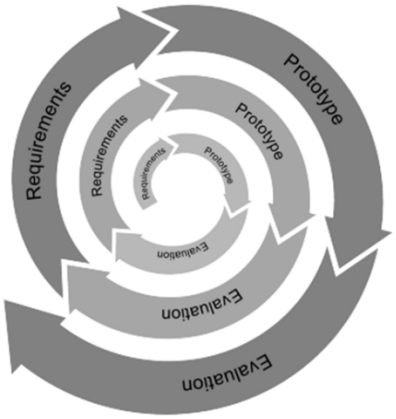
Figure 5. Original proposed spiral model for software development



Source: Author's adaptation according Gujarathi, I. A., Norris, W. R., Patterson, A., & Patterson, A. E. (2024). Spiral Development for Non-Software Product and System Engineering. Authorea Preprints, 1–24..

Another example of a spiral model is shown in Figure 6 proposed by Signoretti et al. and proposes three phases: obtaining the requirements (phase 1), prototype creation (phase 2) and evaluation (phase 3) which are iteratively repeated towards an increasingly refined application (Signoretti et al., 2015).

Figure 6. Spiral model with Iiterative process

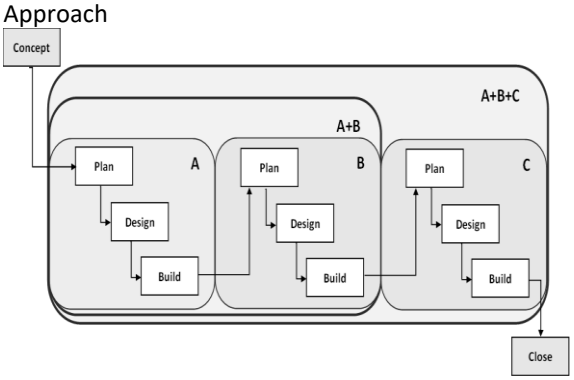


Source: Signoretti, A., Martins, A. I., Almeida, N., Vieira, D., Rosa, A. F., Costa, C. M. M., & Texeira, A. (2015). Trip 4 All: A Gamified App to Provide a New Way to Elderly People to Travel. Procedia Computer Science, 6th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Infoexclusion (DSAI 2015), 67, 301–311.

3. INCREMENTAL PROJECT MANAGEMENT APPROACH

The incremental model of project management involves cyclicality or successive upgrading of the product/service or the expected result. In this process, new functionality or value is added to each previous version of the product or service, i.e. with each new cycle (increment), until the desired end result or fully functional product/service is obtained (Project Management Life Cycles - Evolution over the Years, 2024). This means that "each increment delivers increasing benefit to the user from an early stage and it allows for management of limited resources to cost and timescale constraints." (Gray et al., 2016). So, with the incremental model, different parts of the system are developed at different times and integrated as they are finalized (Cockburn, 2008). From the perspective of the implementation of software projects, for example, in the incremental model, software requirements are divided into several small incremental parts, i.e. modules, so each subsequent module is added to the previous one, building the full functionality of the final software, and each incremental part in some way represents a separate project. This allows early detection of potential problems and modification of the requirements and scope of the project (Jagli & Yeddu, 2017). Another benefit of the incremental project management model is that "incremental delivery reduces the risk of major misalignment between project output and project success" (Mirzaei et al., 2024). In the incremental project management model, one starts with the concept, that is, the overall plan for the entire project, and then, as shown in Figure 7, several iterations of plan, design, and build follow, with "each subsequent build adding functionality to the initial build" (PMI, 2021). Thus, in the figure, the activities in the first increment marked "A" are started; in the next iteration, the activities of the second increment marked "B" are implemented, which is added to the first increment, forming the "A+B" segment, so that finally, the third, last increment marked "C" is added to the functionality of "A+B" finalizing the final result "A+B+C" followed by project closure.

Figure 7. Life Cycle with Incremental Development Approach



Source: Author's adaptation according to PMBOK guide 7th edition

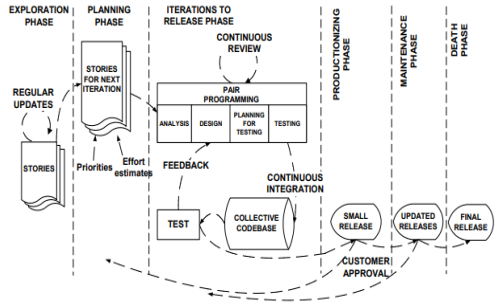
4. AGILE PROJECT MANAGEMENT APPROACH

The application of the agile project management approach began in 2001, although even before this period there were approaches to project management that shared common characteristics with this approach. Namely, in 2001, seventeen world experts in software engineering, calling themselves the "Agile Alliance", formulated and published the so-called "The Agile Manifesto", which consists of 12 principles for software development (Beck et al., 2001). Although initially this methodology was developed for software development, it can be useful in many other areas of project management. The agile methodology is by its nature iterative and incremental at the same time and includes four basic principles: Individuals and Interactions Over Processes and Tools; Working Software Over Comprehensive Documentation; Customer Collaboration Over Contract Negotiation; Responding to Change Over Following a Plan (Beck

et al., 2001). There are a number of different concepts related to agile methodologies such as: Scrum, Extreme Programming (XP), Kanban, Lean, APF (Agile PeopleOps Framework), Feature Driven Development (FDD), Test Driven Development (TDD), Crystal, Dynamic System Development Method (DSDM) (Omonije, 2024; Trivedi, 2013; Vijay Anand & Dinakaran, 2016).

Figure 8 shows the life cycle of XP consisting of five phases: "Exploration" phase in which customers communicate needs they want to be included in the first release; "Planning" Phase in which the priority order of user stories is set and the necessary effort to be undertaken is assessed; Iterations to "Release" Phase includes several iterations of the systems before the first release; "Productionizing" provides for testing and verification of system performance before the system can be released to the customer;" Maintenance" phase includes customer support tasks and "Death" phase occurs when all user stories have been implemented and the system meets customer needs. (Abrahamsson et al., 2017). The characteristic of XP programming, as well as any other agile methodology, is that at the same time, in addition to incrementality, iterativity is followed in accordance with customer requirements and their approval.

Figure 8. life cycle of XP

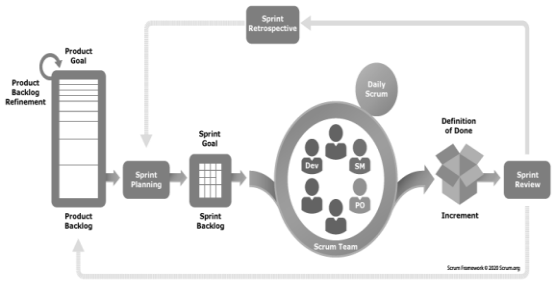


Source: According to Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile Software Development Methods: Review and Analysis. VTT Publication

Another characteristic lifecycle associated with agile methodology is Scrum. Scrum as a methodology is described in detail in the Scrum Guide (first version published in 2010 and the last revised version in 2020) by Ken Schwaber & Jeff Sutherland, although the methodology was developed by them in the early 1990s. The idea of Scrum is for the scrum team (consisting of a Product Owner, a Scrum Master and Developers) to deliver value incrementally (in small pieces at a time) in a collaborative way through iterations that ensure that the final product meets customer expectations (scrum.org, 2025). This methodology includes 19 Scrum processes as part of five Scrum phases: Initiate; Plan and Estimate; Implement; Review and Retrospect; and Release (SCRUMstudy, 2022), in order to create artifacts that are "tangible deliverables produced during software development."(Wagenaar et al., 2015).

Figure 9 visually presents the scrum framework (scrum.org, 2025) whose main components are three scrum artifacts: product backlog, which is a dynamic list of features, requirements, enhancements, and fixes; sprint backlog, i.e. list of items to be completed by the development team in the current sprint cycle; and increment or product from a sprint, together with scrum events that include: sprint panning, sprint, daily scrum, sprint review and sprint retrospective (https://aws.amazon.com, 2025) in order to achieve product goals through iterative processes of product backlog refinement.

Figure 9. Scrum framework



Source: scrum.org

Agile methodologies, due to their nature of incrementality and iterativeness, are particularly useful in cases where there are no clear constraints on the project scope and timeframe, and if "it is difficult to understand the system functionalities during the early phase of the process due to continuous requirements changing" (Angioni et al., 2006).

CONCLUSION

The success of software development projects significantly depends on the choice of an appropriate software development methodology. In this paper, several project management life cycle approaches were considered. Traditional methods such as Predictive project management life cycle approach (Waterfall) offer a well-defined project structure and accurate planning, but on the other hand, this methodology is not suitable for today's turbulent environment in which software projects are developed. That is why these methods often do not meet the requirements imposed by such a dynamic environment.

Unlike these traditional methodologies, adaptive approaches such as Agile project management approach provide flexibility and increased collaboration and thus manage to meet the requirements imposed by a changing environment in which software projects are developed. The characteristics of this model such as: Individuals and Interactions Over Processes and Tools; Working Software Over Comprehensive Documentation; Customer Collaboration Over Contract Negotiation; Responding to Change Over Following a Plan, make this approach the best for application in software development.

However, the choice of a software development model may depend on a number of factors that determine this process. Factors that influence the software development process include the size of the project, the complexity of the project, the goals to be achieved, the time of project implementation, the financial resources for project implementation, etc. Perhaps the best results will be achieved by combining several methodologies depending on the conditions in which the software project is developed and the factors that influence its development.

References:

Aurelia, P. (2014). Business Process Modelling for Projects Costs Management in an Organization. Annals of the „Constantin Brâncuşi" University of Târgu Jiu, Economy Series, Special Issue - Information society and sustainable development

Abdul, R., Khan, H., Gopal, P. R., Yogesh, P. B., Nitin, P. V., Sahebrao, K. S., Sharad, P. N., & Santosh, P. V. (2024). The Waterfall Model-Software Engineering. International Journal of Research Publication and Reviews Journal Homepage: Www.Ijrpr.Com, 5(4), 1–6. www.ijrpr.com

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile Software Development Methods: Review and Analysis. VTT Publication.

Adobe. (2021). Mixing Agile and Waterfall. In Adobe Experience Cloud.

Angioni, M., Carboni, D., Pinna, S., Sanna, R., Serra, N., & Soro, A. (2006). Integrating XP project management in development environments. Journal of Systems Architecture, 52(11), 619–626. https://doi.org/10.1016/J.SYSARC.2006.06.006

APM. (2019). APM Body of Knowledge. In Journal of Lesbian Studies (Vol. 15, Issue 1).

Asana. (2025). Understanding the Iterative Process (with Examples). https://asana.com/resources/iterative-process

Balasubramaniam, V. S., Thumati, P. R. R., Kanchi, P., Agarwal, R., Goel, O., & Shrivastav, E. A. (2023). Research in Engineering Management and Science Evaluating the Impact of Agile and Waterfall Methodologies in Large Scale IT Projects. International Journal of Progressive Research in Engineering Management and Science (IJPREMS), 03(12), 397–412.

Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Manifesto for Agile Software Development. Agile Alliance. https://agilemanifesto.org/

Bell, T. E., & Thayer, T. A. (1976). Software Requirements: Are They Really a Problem? Proceedings of the 2nd International Conference on Software Engineering. IEEE Computer Society Press., 61–68.

Boehm, B. (1986). A spiral model of software development and enhancement. SIGSOFT Softw. Eng. Notes, 11(4), 14–24. https://doi.org/10.1145/12944.12948

Cockburn, A. (2008). Using both incremental and iterative development. CrossTalk, 21(5), 27–30.

Gray, A., Richardson, K., Rooke, K., & Thornburn, T. (2016). Guide to Life Cycles and Life Cycle Models. In INCOSE and the Association for Project Management (Issue 1.1). https://www.apm.org.uk/media/13835/guide-to-lifecycle-models.pdf

Gujarathi, I. A., Norris, W. R., Patterson, A., & Patterson, A. E. (2024). Spiral Development for Non-Software Product and System Engineering. Authorea Preprints, 1–24.

https://aws.amazon.com. (2025). What is Scrum? - Scrum Methodology Explained - AWS. https://aws.amazon.com/what-is/scrum/

Jagli, D., & Yeddu, S. (2017). CloudSDLC: Cloud Software Development Life Cycle. International Journal of Computer Applications, 168(8), 6–10. https://doi.org/10.5120/ijca2017914468

Jenkins, N. (2003). The Iterative Project Model (IPM) - A lightweight methodology for project management. https://www.stickyminds.com/

Kerzner, H. (2009). Project management: a systems approach to planning, scheduling, and controlling. In New York (10th ed.). John Wiley & Sons, Inc.

Labuschagne, C. (2005). Sustainable project life cycle management: Development of social criteria for decision-making.

Lalband, N., & Kavitha, D. (2019). Software engineering for smart healthcare applications. International Journal of Innovative Technology and Exploring Engineering, 8(6 Special Issue 4), 325–331. https://doi.org/10.35940/ijitee.F1066.0486S419

Mirzaei, M., Mabin, V. J., & Zwikael, O. (2024). Customising Hybrid project management methodologies. Production Planning and Control, 0(0), 1–18. https://doi.org/10.1080/09537287.2024.2349231

Omonije, A. (2024). Agile Methodology: A Comprehensive Impact on Modern Business Operations. International Journal of Science and Research (IJSR), 13(2), 132–138. https://doi.org/10.21275/sr24130104148

Petersen, K., Wohlin, C., & Baca, D. (2009). The Waterfall Model in Large-Scale Development. In F. Bomarius, M. Oivo, P. Jaring, & P. Abrahamsson (Eds.), Product-Focused Software Process Improvement (pp. 386–400). Springer Berlin Heidelberg.

PMI. (1996). A Guide to the Project Management Body of Knowledge. Project Management Institute.

PMI. (2017). A guide to the project management body of knowledge / Project Management Institute (PMBOK). In Project Management Institute, Inc. (6th ed.).

PMI. (2021). The standard for project management and a guide to the project management body of knowledge (PMBOK guide). In The Standard for Project Management and A Guide to The Project Management Body of Knowledge (PMBOK guide 7th edition). (Issue July).

Portman, H. (2022). Sensemaking in the Agile Forest - What do iterative and incremental mean in Agile ? PM World Journal, XI(VII), 1–5.

Pressman, R. S. (2010). The Waterfall Model. In Software Engineering: A Practitioner's Approach (Seventh Ed). McGraw-Hill.Project Management Life Cycles - Evolution over the years. (2024). https://www.simplilearn.com/project-management-life-cycles-evolution-article

Radigan, D. (2024). Agile vs. waterfall project management. Atlassian. https://www.atlassian.com/agile/project-management/project-management-intro

scrum.org. (2025). What is Scrum? | Scrum.org. https://www.scrum.org/resources/what-scrum-module

SCRUMstudy. (2022). SCRUM Body of Knowledge (SBOK Guide), Fourth Guide. In SCRUMstudy.

Signoretti, A., Martins, A. I., Almeida, N., Vieira, D., Rosa, A. F., Costa, C. M. M., & Texeira, A. (2015). Trip 4 All: A Gamified App to Provide a New Way to Elderly People to Travel. Procedia Computer Science, 6th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Infoexclusion (DSAI 2015), 67, 301–311. https://doi.org/10.1016/j.procs.2015.09.274

Trivedi, D. (2013). Agile Methodologies. VerisonOne. Agile Made Easier, 48(5), 73–78.

Twproject. (2025). The iterative method within projects: an opportunity to seize - Twproject Blog. https://twproject.com/blog/iterative-method-within-projects-not-alternative-opportunity/

Van Der Merwe, I. W. (2017). How relevant are waterfall project management methodologies in today's modern project environment? July, 1–9. https://doi.org/10.13140/RG.2.2.15971.66083

Vijay Anand, R., & Dinakaran, M. (2016). Popular agile methods in software development: Review and analysis. International Journal of Applied Engineering Research, 11(5), 3433–3437.

Wagenaar, G., Helms, R., Damian, D., & Brinkkemper, S. (2015). Artefacts in agile software development. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9459(March 2017), 133–148. https://doi.org/10.1007/978-3-319-26844-6_10

Wang, T., & Chen, H. M. (2023). Integration of building information modeling and project management in construction project life cycle. Automation in Construction, 150, 104832. https://doi.org/10.1016/J.AUTCON.2023.104832

Watt, A., Barron, M., & Barron, A. (2014). The Project Life Cycle (Phases). In Project Management. BCcampus. https://opentextbc.ca/projectmanagement/chapter/chapter-3-the-project-life-cycle-phases-project-management/