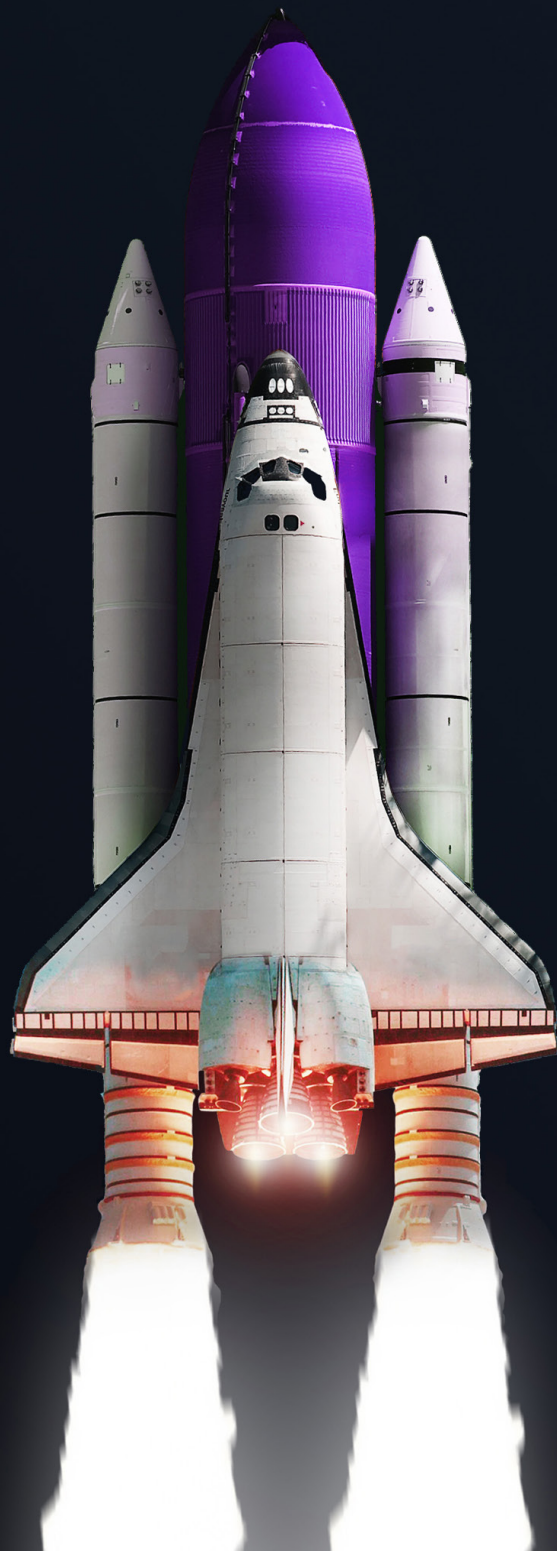




# Delivery Methodology



# Methodology Overview

Delivering complex business systems can be challenging. Studies suggest that the failure rate for large-scale IT business projects is between 70% and 90%. This stems from all sorts of factors such as scope creep, budget issues, inadequate planning, training, competing ideas, poor support, delays, etc. More often than not, you can trace these issues back to a single and often underrated factor: the lack of a clear and comprehensive methodology.

Nexoid's methodology differs from other methodologies because it's not just a best practice document, but a synopsis of decades of experience. Over the years, we have observed the very best and the very worst practices. Our methodology serves as a reservoir of knowledge extracted from clients, suppliers, partners, and of course, our projects. As with everything at Nexoid, we are constantly reviewing and improving our systems and offerings. Perfection is a journey, not a destination.

Our delivery methodology incorporates a hybrid model, utilising the most effective elements from Waterfall, Agile, Lean, and SDLC (System Development Life Cycle). This document is a concise guide on integrating and capitalising on each methodology.

## Contents

**1. Project Discovery:** The project discovery phase employs a bespoke approach, blending traditional SDLC with elements of Waterfall, Agile, and Lean methodologies to precisely meet IT project requirements. Our process starts with a thorough analysis of project needs, engaging in iterative development and client consultations to form a clear and prioritised product backlog. Emphasising the creation of a Minimum Viable Product (MVP) and careful scope management, our methodology is designed to be responsive to changes and aligned with the evolving needs of our clients.

**2. Project initiation:** In the project initiation phase, we focus on comprehensive planning, covering aspects such as personnel involvement, resource allocation, methodology training, and budgeting. Emphasising effective communication and risk management, we establish clear protocols and use Agile methodologies to ensure all stakeholders are aligned and engaged. The culmination of this phase is the creation of a detailed project charter, formalising the project's scope, objectives, and team roles, essential for guiding the project to successful completion.

**3. Software Development Process:** Nexoid's software development approach is a unique blend of Waterfall, Agile, Lean, and SDLC methodologies, tailored to deliver high-quality, client-centric software solutions. Emphasising flexibility, efficiency, and comprehensive oversight, our process begins with a deep understanding of client needs and aims for a Minimum Viable Product within a structured budget framework. This strategic combination allows for responsive development, rigorous planning, and a focus on long-term software viability.

**4. Implementation and Change Management:** The "Implementation and Change Management" phase in a project is crucial, where the focus shifts significantly towards managing the human elements of change. At Nexoid, this phase involves addressing challenges like resource constraints and technology adaptation, while emphasising the importance of clear communication, training, and leadership support. Key to our approach is the identification of "champions" - individuals who advocate for change, provide feedback, and guide others, ensuring a smooth transition and mitigating the risks of project dissatisfaction and failure.



## 5. Post-Deployment Support and Maintenance:

In this phase, we prioritise ensuring software compatibility and functionality amidst the ever-evolving IT landscape. This phase encompasses rigorous testing, continuous updates, user training, and proactive change management, alongside a clear warranty policy for our services. Our comprehensive approach guarantees high-quality maintenance and support, ensuring both immediate effectiveness and long-term reliability of our software solutions.

## 6. Compliance and Security Considerations:

Nexoid's approach to IT compliance and security involves embedding these principles deeply within organisational processes, ensuring not only adherence but continual improvement. Our comprehensive strategy includes proactive monitoring, integrating security into development cycles, collaborative risk management, and a strong focus on continuous improvement through Agile practices. Emphasising GDPR and HIPAA compliance, we maintain a culture of responsibility, ensuring all operations align with evolving legal and industry standards.

**7. Glossary of Terms:** This glossary provides key definitions for terms used in Nexoid's delivery methodology, offering clarity on concepts crucial to the project lifecycle.

# 1. Project Discovery

In the project discovery phase, we begin by understanding what needs to be delivered. We enhance the traditional System Development Life Cycle (SDLC) with a mix of Waterfall, Agile, and Lean practices, customising our approach to meet the specific requirements and expectations of IT projects. This process starts with requirements analysis, involving iterative development and client feedback through direct interviews and consultations.

We create a comprehensive product backlog, a prioritised list of features and functionalities, which forms the foundation of our development work. Sprints, typically lasting 1-2 weeks, are then used to develop, test, and review these features, ensuring we stay ahead in planning and maintain a continuous workflow. This approach allows us to adapt quickly to changes, keeping the development aligned with the client's evolving needs.

Our methodology emphasises the importance of defining the project's objectives and scope, such as business goals, target audience, and the value offered. We engage with stakeholders to identify these elements, ensuring clarity on the product's purpose, users, and interaction. Setting realistic deadlines and defining both functional and non-functional requirements are crucial steps in this process.

We use Agile methodologies to prioritise tasks and focus on developing a Minimum Viable Product (MVP), the most simplified version of the product that still functions solving the business need. This lean approach expedites time to market and lays a foundation for future enhancements. Our methodology is designed to evolve with feedback and real-world use, ensuring that the product not only meets initial requirements but also adapts to emerging needs.

Scope management is key to avoiding scope creep, which can lead to delays and budget increases. We see scope evolution as an opportunity to refine the product, carefully balancing changes to ensure they add value without compromising the project's overall objectives. Determining success criteria, considering interoperability, and assigning clear roles and responsibilities are integral to our process. We also employ a comprehensive User Acceptance Testing (UAT) plan and use the 'Definition of Done' (DoD) to gauge project completion, aligning the metrics of all parties involved.



## 2. Project initiation

In the project initiation phase, we focus on identifying key elements such as necessary personnel, external resources, methodology and project training, budget and scope, as well as roles and responsibilities. Budgeting takes into account all expenses, including manpower, tools, training, and contingency funds. We also emphasise learning from past experiences to guide current projects effectively.

Communication is a crucial aspect at this stage. We establish communication protocols, particularly for feedback, which is central to Agile's iterative process. This involves defining the frequency and channels of communication to keep all team members and stakeholders informed and engaged.

Conducting a preliminary risk assessment is essential to identify potential challenges. We follow a structured approach to risk management, which includes identifying and analysing risks, assigning responsibility, developing mitigation strategies, and regularly monitoring risks. A risk matrix helps us rate risks based on probability and impact, but we remain cautious of its complexity and subjectivity.

Project planning tools are selected for effective scheduling, budgeting, resource allocation, and communication. The Risk Register is another tool we employ to track and mitigate identified risks.

In terms of project costing in Agile environments, we estimate effort through user stories, assigning story points and running initial sprints to gauge effort levels. The total cost is calculated by considering sprint costs and adding a contingency margin for unexpected issues. We offer pricing options that can range from fixed estimates to a pay-as-you-go model.

We also use a burndown rate to monitor the progress of the project, ensuring that work is completed within the set timeframe and identifying any deviations from the expected trajectory.

Finally, a comprehensive project charter is developed, outlining the project's vision, objectives, scope, and team composition. This document, especially crucial for larger projects, provides a formal authorisation for the project and is typically approved by the project sponsor, ensuring all stakeholders are aligned and the project is set for success.

## 3. Software Development Process

In the world of project management, Waterfall and Agile stand out as two predominant styles. Waterfall, with its structured approach, is excellent for budgeting and planning, but it's often chosen more out of habit than suitability for software projects. In contrast, Nexoid prefers an Agile or evolutionary methodology, which is widely recognized as the standard in software development. Agile's iterative nature, despite potential challenges with budget overruns and shifting objectives, enhances project success rates.

At Nexoid, we've developed a unique approach that blends the strengths of Waterfall, Agile, Lean, and SDLC (System Development Life Cycle) methodologies. This combination ensures we deliver high-quality software solutions that are precisely tailored to our clients' needs.

For strategic planning and budgeting, we use Waterfall due to its sequential nature, which aligns well with high-level management expectations and provides a clear project roadmap. However, we're cautious of its inflexibility, as it may not cater well to evolving business needs.

Agile methodology comes into play for its adaptability and focus on client value. It allows us to be responsive in product development and encourages regular feedback from clients. Our project managers use strict prioritisation and review processes to avoid the potential sprawl of Agile projects.



We further enhance our approach with Lean principles, aiming for maximum targeted effort. By focusing on creating a Minimum Viable Product (MVP), we deliver what's needed to solve the customer's problem without over-engineering, keeping our projects focused and on budget. We ensure that Lean's rigorous, evidence-based approach is consistently applied.

SDLC is our comprehensive framework that goes beyond the initial software launch. It covers the entire lifespan of the software, emphasising post-launch activities like support, maintenance, and iterative improvements, thus ensuring the software's adaptability and longevity.

Our development process starts with a deep understanding of the client's needs, involving discussions with clients, developers, and business analysts. This initial understanding aids in budget estimation and defining the MVP, which is the quickest solution that meets the customer's technical needs.

Our aim is to reach MVP within 40% of the allocated budget, leaving the remaining 60% for further improvements and refinements. This latter portion is managed using typical Agile methodologies, where we demonstrate the MVP to the client and create a backlog - a list of changes, improvements, and additions, each with an estimated time for completion.

We then move into fortnightly sprints, prioritising tasks for the next two weeks. This iterative sprint pattern ensures we make the most efficient use of resources and maximise value for our clients.

## 4. Implementation and Change Management

The "Implementation and Change Management" phase is a pivotal stage in a project where its success is heavily influenced by how well the human aspects are managed. Despite thorough planning, various challenges can arise, such as resource limitations or

technological issues. However, the readiness and willingness of the people involved to adapt to changes play a crucial role.

This phase demands more than just a robust plan; it requires careful attention to the individuals involved. Clear communication, effective training, and supportive leadership are key to navigating this stage successfully. Ensuring that everyone understands and embraces the new changes is vital. Resistance to change is a common hurdle, as people are often hesitant to alter familiar ways of working. Recognising that successful change management involves both the adoption of new systems and the willingness of people to engage with these changes is crucial.

At Nexoid, we address this challenge by identifying and nurturing "champions" within the team. These individuals are characterised by their positive attitude, eagerness to learn, and desire to advance their careers. Champions play multiple roles: they advocate for the new changes, ensuring everyone understands their importance and benefits; they listen to the team's concerns and feedback, facilitating timely resolutions to issues; and they act as guides, helping others to navigate and embrace the new processes.

Champions are also instrumental during the release phase of a project. Software releases with faults can cause dissatisfaction within the team, potentially leading to a cascading effect of negativity and project failure. By having champions in place, this risk is mitigated as they help maintain team morale and focus on the positive aspects of the project.

While champions are a valuable asset, it's important to note that this approach does not directly address the quality of work or the actual progress of the remaining tasks. Therefore, we use this strategy as one of several tools within the project, ensuring a balanced and comprehensive approach to implementation and change management.



## 5. Post-Deployment Support and Maintenance

In the “Post-Deployment Support and Maintenance” stage of software development, we recognise that software release is complex, requiring numerous factors and components to work in harmony. As the IT world is constantly evolving, our ERP and ITSM solutions require continuous updates to stay compatible with new technological advancements. Post-deployment support and maintenance are integral to our service, ensuring our solutions are compatible with both current and future technology.

We employ techniques like Feature Flags and A/B testing to manage functionality and gather feedback, ensuring software readiness before widespread release. Our testing phases, including unit, integration, system, and User Acceptance Testing (UAT), are rigorous to ensure compliance with technical standards and requirements.

Once software goes live, our focus shifts to maintenance and continuous improvement, including routine updates and adjustments. We prioritise user training and support to ensure proficiency and comfort with the new software. Our maintenance strategy involves regular health checks, upgrades, and performance tuning, all tailored to the needs of users and administrators.

We proactively manage software changes, with clear protocols for change implementation and thorough impact analyses. Our approach ensures that software remains reliable, effective, and aligns with evolving user requirements.

Our Definition of Done (DOD) criteria means a project is only finished when all deliverables meet stakeholder approval, acceptance criteria, and the agreed scope. We ensure that all commitments, including verbal promises, are fulfilled.

We offer a clear warranty period for our services, covering free maintenance support for a set time after release. This warranty applies to both custom client-specific code and our shared Nexoid platform, ensuring seamless updates and fixes. However, we distinguish between changes affecting our shared platform and client-specific systems, with maintenance responsibilities clearly defined.

By adhering to this methodology, we aim for the highest possible success rate, ensuring satisfaction at every level. A comprehensive post-project report is compiled and archived for future reference, leaving no room for interpretation and ensuring clarity and success in our software development endeavours.

## 6. Compliance and Security Considerations

In Nexoid’s dynamic IT landscape, compliance and security are embedded into organisational processes, ensuring active maintenance and improvement of standards. We go beyond basic client and industry requirements by engaging in robust monitoring of safety, risk, and compliance, including regular assessments, real-time surveillance, and stakeholder engagement. Our approach to safety and legal compliance is rigorous, integrating security practices into the development cycle and resolving compliance conflicts effectively.

Risk management is collaborative, involving clients in identifying and mitigating risks, with warranties underlining our commitment to managing these risks. Our methodology for continuous improvement is inspired by the scientific method, involving identification, planning, testing, and refining processes. This is facilitated through Agile management, enabling us to adapt and improve continuously.

We place a strong emphasis on feedback loops, training, and leveraging technology to enhance compliance measures. Our training initiatives are tailored to client needs, ensuring awareness of compliance standards and security best practices. Compliance monitoring



is comprehensive, involving regular audits, legal updates, and cross-functional team collaboration to ensure adherence at every level.

Particularly, we focus on GDPR, HIPAA, and other data protection regulations, ensuring alignment of all IT operations with these laws. Our GDPR compliance framework includes Data Protection Impact Assessments, maintaining detailed data processing records, and appointing a Data Protection Officer.

In summary, Nexoid's commitment to compliance and security is a culture of responsibility and awareness, combining stringent monitoring, client-participative risk management, continuous improvement, and comprehensive training to embed a deep understanding of compliance requirements in all operations.

## 7. Glossary of Terms

**A/B Testing:** A method of comparing two versions of a webpage or app against each other to determine which one performs better.

**Agile:** A project management methodology characterised by the division of tasks into short phases or iterations (sprints) and frequent reassessment and adaptation of plans.

**Backlog:** A prioritised list of tasks and requirements that the project team maintains for a project.

**Burndown Rate:** A metric in Agile project management that shows the rate at which work is completed and how much work remains.

**Champion:** A person who supports, promotes, and helps drive the change within an organisation, playing a key role in the successful implementation of new systems or processes.

**Change Management:** A systematic approach to dealing with change, both from the perspective of an organisation and the individual.

**Continuous Improvement:** The ongoing effort to improve products, services, or processes. These efforts can seek "incremental"

improvement over time or "breakthrough" improvement all at once.

**Cost Risks:** Risks associated with the possibility of cost overruns due to inaccurate estimations or budgeting.

**Definition of Done (DoD):** A clear and concise list of criteria which all project tasks must meet before they are considered complete.

**DevOps / DevSecOps:** An approach to culture, automation, and platform design that integrates security as a shared responsibility throughout the entire IT lifecycle.

**External Risks:** Risks that come from outside the project's direct control, such as regulatory changes or issues with vendors.

**Feature Flags:** A technique in software development that allows teams to modify system behaviour without changing code.

**Feedback Loop:** A process in which the outputs of a system are circled back and used as inputs.

**Financial Risks:** These encompass risks stemming from financial uncertainties like budget inaccuracies or funding inconsistencies.

**Maintenance:** The process of maintaining the software after it goes live, including routine updates and fixes.

**Minimum Viable Product (MVP):** The most pared-down version of a product that can still be released to market with a minimum set of features to satisfy early adopters and provide feedback for future development.

**Mitigation Strategies:** Techniques used to manage risks, including avoidance, acceptance, reduction, transfer, exploitation, enhancement, and sharing.

**Module:** In Nexoid ERP, a module is a distinct part of the software that can be used independently or with other modules to perform a specific business function.

**Performance Risks:** Risks that the project's deliverables may not meet the required or expected standards or benchmarks.



**Positive Risks:** Also known as opportunities; these are events that, if they occur, will have a beneficial impact on the project's objectives, like finishing tasks ahead of schedule or under budget.

**Post-Deployment Support:** Activities that occur after the software is released to ensure its continued operation and maintenance.

**Project Lifecycle:** The stages through which a project passes, typically initiation, planning, execution, monitoring, and closure.

**Project Risk:** The potential for unforeseen challenges or threats that may negatively influence the successful completion of a project. Risks can be uncertain and may not always materialise.

**Release:** The distribution of the final or new incremental version of the software to the users.

**Risk Management Process:** A systematic approach to identifying, analysing, and responding to project risks, which includes minimising, monitoring, and controlling the probability or impact of unfortunate events.

**Risk Matrix:** A grid for plotting the probability of a risk against its impact to help in the prioritisation of risk management actions.

**Risk Register:** A tool used in risk management processes to track identified risks, their severity, and the actions steps or plans put in place to mitigate them.

**Schedule Risks:** These include risks related to project delays or extensions beyond the planned schedule, often exacerbated by scope creep.

**Scope Creep:** The uncontrolled expansion to project scope without adjustments to time, cost, and resources.

**SMART Goals:** Acronym for Specific, Measurable, Attainable, Realistic, and Timely, which are the criteria for setting objectives for a project.

**Sprint:** A set timeframe during which specific work has to be completed and made ready for review in an Agile framework.

**Stakeholder:** An individual, group, or organisation who may affect, be affected by, or perceive itself to be affected by a decision, activity, or outcome of a project.

**Strategic Risks:** Risks arising from strategic mistakes, employing outdated methodologies, or incompatibilities with company culture.

**System Development Life Cycle (SDLC):** A process for planning, creating, testing, and deploying an information system, with stages often including analysis, design, development, and implementation.

**System Testing:** An integrated testing phase where the complete system is tested to verify that it meets the specified requirements.

**UAT (User Acceptance Testing):** A phase in the software development lifecycle where the software is tested by the intended audience or stakeholders in conditions that simulate production, to ensure it meets the required specifications and functionality.

**UAT Environment:** A testing environment that closely resembles the production environment but is used exclusively for user acceptance testing to safeguard against untested changes directly affecting the live system.

**Workflow:** The sequence of processes through which a piece of work passes from initiation to completion within the Nexoid ERP system.

