Nexelus Development Process

SOC 1 Type I Document

esm_logo.jpg

Nexelus USA  
New York, NEW YORK (NY)  
646-558-1950 ext.128

**6/21/2022**

# Document Information

The following table shows the details for document creation, review, approval, and effective date.

| **Category** | **Information** |
| --- | --- |
| Work Product: | Nexelus Development Process – SOC 2 Document |
| Project Name: | Development, Development Process |
| Function Name: | SOC 2 Document |
| Version: | 1.0 |
| Status: | Draft |
| Author(s): | Tauseef Shahzad |
| Reviewer(s): | Asim Jameel |
| Approver(s): | Imran Rahman |
| Control Status: | CONTROLLED, PROTECTED |
| Disclaimer: | This document contains confidential information. Do not distribute this document without prior approval from Nexelus. |

# Revision History

The following table is used for revision details of this document.

| **Author(s)** | **Date** | **Version** | **Description of Change** |
| --- | --- | --- | --- |
| Tauseef Shahzad | October 10, 2021 | 1\_0 | Initial Draft |
| Tauseef Shahzad | October 26, 2022 | 1\_1 | Added Hybrid Development Model description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

Contents

[Document Information ii](#_Toc113004066)

[Revision History iii](#_Toc113004067)

[Table of Contents i](#_Toc113004068)

[Scope 1](#_Toc113004069)

[Development Lifecycle Models 2](#_Toc113004070)

[Waterfall Model 3](#_Toc113004071)

[Waterfall Model – Design 3](#_Toc113004072)

[Waterfall Model – Application 4](#_Toc113004073)

[RAD Model 5](#_Toc113004074)

[RAD model has the following phases: 5](#_Toc113004075)

[Advantages and Disadvantages of RAD 5](#_Toc113004076)

[Incremental Model 6](#_Toc113004077)

[Agile Methodology 8](#_Toc113004078)

[Benefits of Agile Development 8](#_Toc113004079)

[Problems with Traditional Development 9](#_Toc113004080)

[Suitability of Agile Methods 10](#_Toc113004081)

[Hybrid Methodologies 12](#_Toc113004082)

[The V-Model 12](#_Toc113004083)

[The Spiral Model 13](#_Toc113004084)

[Design, develop and test the increment 13](#_Toc113004085)

[Iterative and Incremental Development 13](#_Toc113004086)

[Nexelus SDLC Model 15](#_Toc113004087)

[Project Management 16](#_Toc113004088)

[Scope 17](#_Toc113004089)

[Policy 18](#_Toc113004090)

[Roles and Responsibilities 19](#_Toc113004091)

[Customer/Client 19](#_Toc113004092)

[Principal Software Architect 19](#_Toc113004093)

[CEO 19](#_Toc113004094)

[General Manager Development 19](#_Toc113004095)

[Development Lead 20](#_Toc113004096)

[QA Lead 20](#_Toc113004097)

[Development Team 21](#_Toc113004098)

[Deployment Lead 21](#_Toc113004099)

[Technical Writer 21](#_Toc113004100)

[Project/Release Management Process 22](#_Toc113004101)

[Project Initiation 22](#_Toc113004102)

[Design 22](#_Toc113004103)

[Project Discovery 22](#_Toc113004104)

[High Level Project Plan 22](#_Toc113004105)

[Project Approval 23](#_Toc113004106)

[Project Kickoff Meeting 23](#_Toc113004107)

[Detailed Project Plan 23](#_Toc113004108)

[Develop Detail Schedule 23](#_Toc113004109)

[Plan Resources 24](#_Toc113004110)

[Plan Monitoring and Control 24](#_Toc113004111)

[Project Execution 25](#_Toc113004112)

[Project Release Cycle 25](#_Toc113004113)

[Conduct Monitoring and Control 26](#_Toc113004114)

[Project Closure 27](#_Toc113004115)

[Release Management Process 28](#_Toc113004116)

[Scope 28](#_Toc113004117)

[Roles and Responsibilities 29](#_Toc113004118)

[Release Manager 29](#_Toc113004119)

[QA Lead/QA Engineer 29](#_Toc113004120)

[DB Lead/Database Developer 29](#_Toc113004121)

[Development Lead/Software Engineer 29](#_Toc113004122)

[Process 30](#_Toc113004123)

[Change Management 32](#_Toc113004124)

[Change Management Process 33](#_Toc113004125)

[Request for Change 33](#_Toc113004126)

[Impact Analysis and System Design Specifications 33](#_Toc113004127)

[Implement Change 34](#_Toc113004128)

[Review / Reporting 34](#_Toc113004129)

[Activity Diagram 35](#_Toc113004130)

# Scope

The scope of this document is to define and establish Software Development process and practices implemented organization wide. All software development personnel must comply with the process defined in this document.

Secure Software Development Lifecycle Standards are developed and implemented with access to program source code restricted based on principle of least privilege. For applications that store or transmit confidential information, security controls are implemented to limit output to minimum necessary as defined by the user. All newly developed software and updates or revisions to existing software are fully tested and accepted prior to deployment to the production environment.

**Reference**

SSAE-18 SOC 1 Type II – Requirements

# Development Lifecycle Models

Traditional approach includes following SDLC model:

## Waterfall Model

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

### Waterfall Model – Design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.



The sequential phases in Waterfall model are −

#### Requirement Gathering and analysis

All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

#### System Design

The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

#### Implementation

With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

#### Integration and Testing

All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

#### Deployment of system

Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

#### Maintenance

There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

### Waterfall Model – Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

* Requirements are very well documented, clear and fixed.
* Product definition is stable.
* Technology is understood and is not dynamic.
* There are no ambiguous requirements.
* Ample resources with required expertise are available to support the product.
* The project is short.

## RAD Model

RAD is a linear sequential Software Development Process model that emphasis an extremely short development cycle using a component-based construction approach. If the requirements are well understood and well defined, and the project scope is constrained, the RAD process enables a development team to create a fully functional system with in very short time period.

### RAD model has the following phases:

#### Business Modeling

The Information Flow among business functions is defined by answering questions like what information drives the business process, what information is generated, who generates it, where does the information go, who processes it and so on.

#### Data Modeling

The information collected from Business Modeling is refined into a set of data objects (entities) that are needed to support the business. The attributes (character of each entity) are identified and the relation between these data objects (entities) is defined.

#### Process Modeling

The data object defined in the Data Modeling Phase are transformed to achieve the information flow necessary to implement a business function. Processing descriptions are created for adding, modifying, deleting or retrieving a data object.

#### Application Generation

Automated tools are used to facilitate construction of the software; even they use the 4th GL techniques.

#### Testing and Turn over

Many of the programming components have already been tested since RAD emphasis reuse. This reduces overall testing time. New components, however, must be tested and all interfaces must be fully exercised.

### Advantages and Disadvantages of RAD

RAD reduces the development time and reusability of components to help to speed up development. All functions are modularized so it is easy to work with.

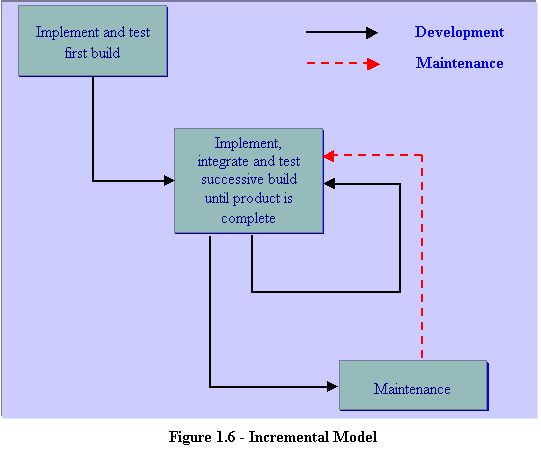
For large projects RAD requires highly skilled engineers in the team. Both end customer and developer should be committed to complete the system in an abbreviated time frame. If there is a lack of commitment, RAD will fail. RAD is based on Object Oriented approach and if it is difficult to modularize the project the RAD may not work well.

## Incremental Model

The model is designed, implemented and tested as a series of incremental builds until the product is finished. A build consists of pieces of code from various modules that interact together to provide a specific function.

At each stage of the IM a new build is coded and then integrated into the structure, which is tested as a whole. Note that the product is only deemed as completed when it satisfies all of its requirements.

Incremental development is a scheduling and staging strategy in which the various parts of the system are developed at different times or rates, and integrated as they are completed. It does not imply, require nor preclude iterative development or waterfall development - both of those are rework strategies. The alternative to incremental development is to develop the entire system with ‘big bang’ integration.



This model combines the elements of the waterfall model with the iterative philosophy of prototyping. However, unlike prototyping the IM focuses on the delivery of an operational product at the end of each increment.

An example of this incremental approach is observed in the development of word processing applications where the following services are provided on subsequent builds:

• Basic file management, editing and document production functions

• Advanced editing and document production functions

• Spell and grammar checking

• Advance page layout

The first increment is usually the core product which addresses the basic requirements of the system. This maybe either be used by the client or subjected to detailed review to develop a plan for the next increment. This plan addresses the modification of the core product to better meet the needs of the customer, and the delivery of additionally functionality. More specifically, at each stage

• The client assigns a value to each build not yet implemented

• The developer estimates cost of developing each build

• The resulting value-to-cost ratio is the criterion used for selecting which build is delivered next

Essentially the build with the highest value-to-cost ratio is the one that provides the client with the most functionality (value) for the least cost. Using this method the client has a usable product at all of the development stages.

## Agile Methodology

Agile methods emphasize working software as the primary measure of progress. Combined with the preference for face-to-face communication, agile methods usually produce less written documentation than other methods. In an agile project; documentation, Gantt charts and other project artifacts all rank equally with working product. However, when stakeholders are asked to priorities deliverables for demonstration at the end of the current iteration, they generally prefer to see working product. Stakeholders are encouraged to priorities iteration outcomes based exclusively on business value perceived at the beginning of the iteration. If documentation represents higher business value than working software in any particular iteration then stakeholders give it a higher priority than working software. The (cross-functional) development team will in accord, produce that documentation instead of lower priority software.

Agile means ‘being able to quickly change direction’. In software development, it requires strong discipline to code for agility. It includes writing tests for functionality before coding. It calls for naming of functionality to exactly match the intent and the terminology of the problem domain. It demands cessation of coding when the tests are passed. The sum total of all the disciplines delivers an ability to change direction quickly. New and unexpected functionality required to cope with a sudden change in the business landscape can be inserted in existing code using test-driven development and all the previous tests will pass or fail to instantly indicate where code needs to be refactored to stay functional. If functionality is added before it is required then it becomes "dead weight" when refactoring is called for.

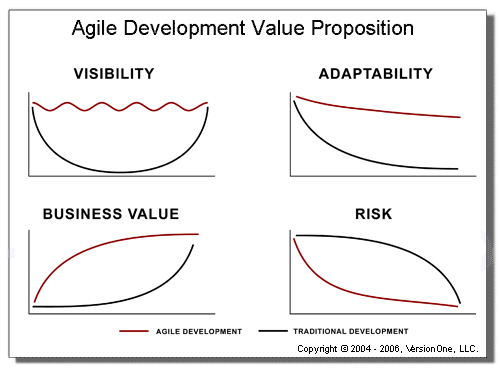
### Benefits of Agile Development

Agile methods grew out of the real-life project experiences of leading software professionals who had experienced the challenges and limitations of traditional waterfall development on project after project. The approach promoted by agile development is in direct response to the issue associated with traditional software development – both in terms of overall philosophy as well as specific processes.

Agile development, in its simplest form, offers a lightweight framework for helping teams, given a constantly evolving functional and technical landscape, maintain a focus on the rapid delivery of business value (i.e., “bang for the buck”). As a result of this focus and its associated benefits, organizations are capable of significantly reducing the overall risk associated with software development.

In particular, agile development accelerates the delivery of initial business value, and through a process of continuous planning and feedback, it is able to ensure that value is continuing to be maximized throughout the development process. As a result of this iterative planning and feedback loop, teams are able to continuously align the delivered software with desired business needs, easily adapting to changing requirements throughout the process. By measuring and evaluating status based on the undeniable truth of working, testing software, much more accurate visibility into the actual progress of projects is available. As a result of following an agile process, at the conclusion of a project is a software system that much better addresses the business and customer needs.

The diagram below displays the differences between agile and waterfall development processes. By delivering working, tested, deployable software on an incremental basis, agile development delivers increased value, visibility, and adaptability much earlier in the lifecycle, significantly reducing project risk.



1. Agile Development Value Proposition

### Problems with Traditional Development

According to the Standish Group's famous CHAOS Report of 2000, 25% of all projects still fail outright through eventual cancellation, with no useful software deployed. This represents a big improvement over previous CHAOS reports from past years. And now there is more evidence of the same kind. In agile and Iterative Development: a Manager's Guide, renowned consultant and author Craig Larman does a thorough job of debunking the traditional waterfall model once and for all.

The numbers are overwhelming. A study in the United Kingdom shows that of 1,027 projects, only 13% did not fail, and waterfall-style scope management was the "single largest contributing factor for failure, being cited in 82% of the projects as the number one problem." A 1995 study of over $37 billion USD worth of US Defense Department projects concluded that "46% of the systems so egregiously did not meet the real needs (although they met the specifications) that they were never successfully used, and another 20% required extensive rework" to be usable.

Larman also points that in "another study of 6,700 projects, it was found that four out of the five key factors contributing to project failure were associated with and aggravated by the waterfall model, including inability to deal with changing requirements, and problems with late integration." And finally, another study of over 400 waterfall projects reported that only 10% of the developed code was actually deployed, and of that, only 20% was actually used.

These numbers reinforce what many of us have experienced personally: the waterfall approach is a risky and expensive way to build software systems. This is the real reason why the much of industry is investigating and/or implementing agile alternatives. Some of the principles behind the agile Manifesto are:

• Customer satisfaction by rapid, continuous delivery of useful software

• Working software is delivered frequently (weeks rather than months)

• Working software is the principal measure of progress

• Even late changes in requirements are welcomed

• Close, daily cooperation between business people and developers

• Face-to-face conversation is the best form of communication (Co-location)

• Projects are built around motivated individuals, who should be trusted

• Continuous attention to technical excellence and good design

• Simplicity

• Self-organizing teams

• Regular adaptation to changing circumstances

### Suitability of Agile Methods

There is little if any consensus on what types of software projects are best suited for agile methodologies. Many large organizations have difficulty bridging the gap between a more traditional waterfall method and an agile one.

Large scale agile software development remains an active research area. Agile development has been widely documented as working well for small co-located teams. Some things that can negatively impact the success of an agile project are:

• Large scale development efforts (>20 developers), though scaling strategies and evidence to the contrary have been described.

• Distributed development efforts (non-co-located teams). Strategies have been described in Bridging the Distance and Using an agile Software Process with Offshore Development

• Command-and-control company cultures

• Forcing an agile process on a development team

Several successful large scale agile projects have been documented. BT has had several hundred developers situated in the UK, Ireland and India working collaboratively on projects and using agile methods. While questions still arise about the suitability of some agile methods to certain project types, it would appear that scale or geography, are not necessarily barriers to success.

Barry Boehm and Richard Turner suggest that risk analysis be used to choose between adaptive ("agile") and predictive ("plan-driven") methods. The authors suggest that each side of the continuum has its own home ground as follows:

Agile home ground:

• Low criticality

• Senior developers

• Requirements change very often

• Small number of developers

• Culture that thrives on chaos

Plan-driven home ground:

• High criticality

• Junior developers

• Requirements don't change too often

• Large number of developers

• Culture that demands order

## Hybrid Methodologies

There are many different hybrid methodologies that attempt to apply feedback mechanisms to the traditional waterfall model so that technical and functional shortcomings in the original design that are discovered during development can be more quickly incorporated. Some of the more popular hybrid models include:

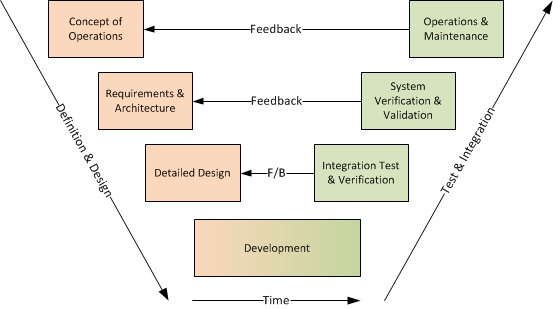
V-Model

Spiral Model

Iterative and Incremental

### The V-Model

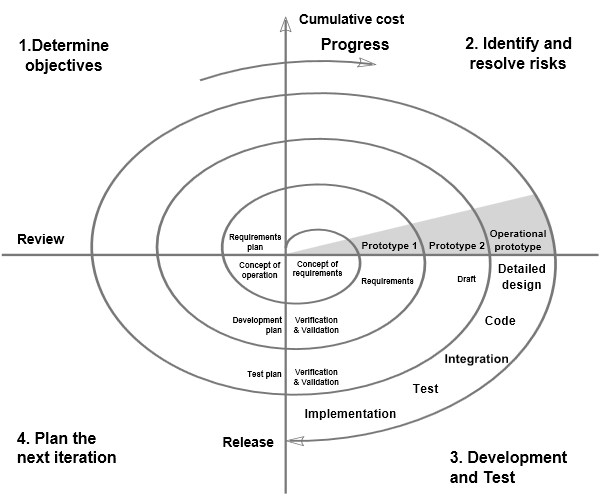
The V-Model of software development uses a modified waterfall to provide a sequential development methodology that has feedback mechanisms between the pre-development and post-development phases of the lifecycle:



Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape. The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing. The horizontal and vertical axes represents time or project completeness (left-to-right) and level of abstraction (coarsest-grain abstraction uppermost), respectively.

### The Spiral Model

The spiral model is a risk-driven process model generator for software projects. Based on the unique risk patterns of a given project, the spiral model guides a team to adopt elements of one or more process models, such as incremental, waterfall, or evolutionary prototyping.



The key distinguishing characteristics of the Spiral Model are that each iteration of the system follows the four key phases that are designed to identify and mitigate risks:

Determining the objectives, planning the scope of the increment

Prototyping, experimentation and research to identify and resolve potential risks (technical, conceptual, etc.)

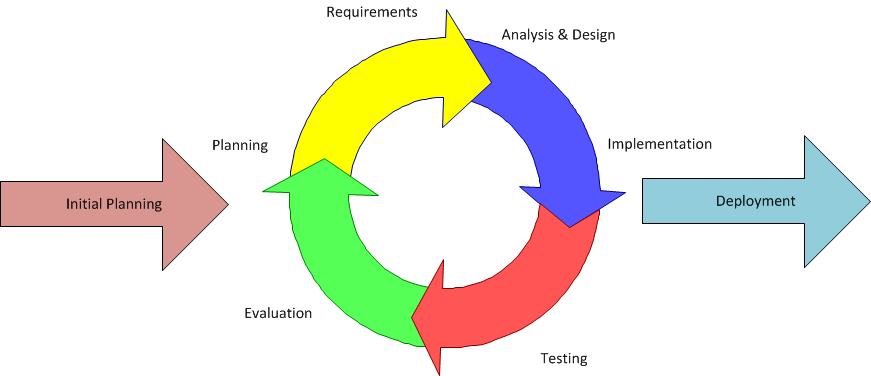
### Design, develop and test the increment

Release and monitor the increment, and use feedback to aid in planning the next iteration

The iterative nature of the Spiral model makes it an early example of a hybrid waterfall-agile methodology and follows many characteristics (prototypes, experiments/spike solutions) that exist in other more recent pure Agile methodologies such as Scrum, XP and AUP.

### Iterative and Incremental Development

Iterative and Incremental development is any combination of both iterative design or iterative method and incremental build model for software development:



The use of iterative/incremental development attempts to mitigate the main criticisms of waterfall / sequential development methodologies since the entire project is broken down into smaller increments or iterations that can apply the lessons learned from the previous iteration.

Learning comes from both the development and use of the system, where possible key steps in the process start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving versions until the full system is implemented. At each iteration, design modifications are made, and new functional capabilities are added.

Although iterative development may look at first glance like agile development, there are several key differences, the main one being that iterative development generally follows the same steps as waterfall, just happening in smaller units of time and that with agile you have, a potentially shippable product, at all times, whereas that is not necessarily the case with iterative development.

## Nexelus SDLC Model

At Nexelus Iterative and Incremental Development of Hybrid SDLC model is used to facilitate release management as per decision taken by the respective project managers.

However, any other development model may be adopted if required for a particular build or project with prior formal approval of CEO and/or Product Owner.

# Project Management

Project management is a set of activities which includes initiating, planning, executing, controlling and closing projects. This document provides information on Nexelus process for project management. Following sections provide details of purpose, policy and scope of project management process.

The purpose of Project Management Process Guide is to establish guidelines for developing realistic plans for project activities in order to deliver successful projects across the organization. The guide covers all the project related activities including Project Initiation, Planning, Design, Implementation, Monitor & Control and Closeout all Software Projects produced by Nexelus.

## Scope

The scope of this document is limited to the project management process. The project management process is initiated when a project has been awarded. It covers the overall management of the project starting from project initiation to its closure. The process includes preliminary activities to initiate the project, establishing the project scope, determining the project lifecycle, estimating level of effort and associated cost, deriving the work break down structure, and establishing project schedule and budget. Other planning activities include data management, resource allocation and skill management.

## Policy

Nexelus performs Project Management (PM) activities and its related processes whenever a new project or product release is initiated, or any change suggested by client is incorporated. Project management activities are performed by respective roles including Principal Software Architect, GM Development, QA lead and development team lead.

The resources involved in PM activities are committed to follow the following principles:

* Requirements should be finalized and approved from client. There should be a clear understanding of specifications between all stakeholders.
* There should be complete information on the scope of work and technical approach on the proposed project in project proposal.
* Initial project information should be described properly to all project stakeholders during project kickoff meeting.
* Project size unit (PSU) should be estimated properly using top level WBS.
* Top level effort should be estimated, and major milestone should be set carefully.
* Detailed level effort should be estimated carefully for each of the following phases of the project:
  + Discovery
  + Development
  + Testing
  + Releases
  + Deployment
* Project monitoring and milestone review should be properly conducted from Discovery to Releases phases properly.
* Analysis of the complete project should be performed, and relevant documents should be updated properly on project closure.

## Roles and Responsibilities

This section provides information on the different roles involved in the project management process and defines responsibilities for each role.

### Customer/Client

A Customer’s role during project is crucial to the success of the project. One of the most important tasks during a project is, to keep the customer/client informed and engaged with everything that’s happening for the project. It is imperative that the customer is involved in the following activities:

* Information gathering for requirements.
* Stakeholder involvement meetings.
* Project status information sharing.
* Project deliverables confirmation and approval.

### Principal Software Architect

Principal Software Architect compiles customer requirements as well as future product features enhancements with consultation with CEO. He also creates following documents:

* Feature list planned for next release
* Requirement Analysis
* Prepare design specifications
* Perform Initial R&D for new features
* Monitor development progress in collaboration with General Manager Development
* Review client feedback and take necessary actions.
* Acts as primary liaison between client, higher management, and development team.

### CEO

CEO coordinates with Client, and Principal Software architect and prioritize the feature list for new release. CEO also reviews the technical specifications before initiating development work.

### General Manager Development

General Manager Development is responsible for developing and/or reviewing the relevant work products. The GM Development may be involved in following activities during Project Management process:

* Create Work Breakdown Structure
* Create Project Estimates.
* Monitor the project development activities.
* Monitor the project verification and validation activities.
* Manage deliverables according to the project plan.
* Create Project Repositories, Release Backlogs and performs code merging on TFS.
* Perform monitoring and control activities for detailed level project planning.
* Evaluate the success of project against its targets.
* Collaborate with development and QA teams
* Ensure timely delivery of the product
* Coordinate with US Team on development progress
* Review the risks and their contingency and mitigation strategies.
* Assign resources to respective teams or projects.

### Development Lead

The Development Lead plays a primary role in the project and is responsible for successful completion of the project. Development Lead is responsible for developing and managing the relevant work products. The Project Manager may be involved in following activities during Project Management process:

* Modify/review System Design Specification/ User Requirements Specification
* Assign tasks to project team members.
* Monitor the project releases activities.
* Compile and update the Risk Management Workbook with collaboration with QA-Lead.
* Perform peer reviews of product components and product.
* Monitor data summary workbook compilation and verification.
* Manages project integration activities.

### QA Lead

QA Lead is responsible for developing and managing the relevant work products. The QA Lead may be involved in following activities during Project Management process:

* Modify/Review System Design Specification/ User Requirements Specification.
* Modify/review Requirements Management Workbook.
* Compile and update the Risk Management Workbook with collaboration with Development-Lead.
* Maintain the quality of the system as per the specification and business requirement.
* Managing QA resourcing allocation and involvement.
* Monitor the project testing, verification, validation activities.
* Serves as a resource and communication point for decisions about test schedule changes.
* Develop/modify integration and deployment plans.

### Development Team

Development team is responsible for the actual development work of the project. Every team member is involved in following activities with respect to Project Management process:

* Provide feedback for System Design Specification/ User Requirements Specification.
* Provide feedback for detailed level WBS.
* Integrate software components and third-party programs. Involved in coding, unit testing and bug fixing activities.
* Participate in status review and milestone meetings.
* Involved in fixing issues raised during verification, validation testing and peer reviews.
* Involved in estimate and schedule of project
* Involved in data collection activities.
* Involved in documentation and reporting.

### Deployment Lead

Deployment Lead is responsible for all deployments on client test and production environments. Other teams do not have access or rights to perform deployment on these environments.

### Technical Writer

Technical writer is responsible for release notes and user guides for using Nexelus product and features.

## Project/Release Management Process

This section provides information on overall project and release management process at Nexelus. It serves as a process guide that outlines the main activities that should be considered and/or practiced during a project execution at the organization. Normally each product release is considered a new project. Following are the activities for Project Management Process:

### Project Initiation

Product enhancement and change requests are compiled by Principal Software Architect. A list of new features and enhancements is also compiled with consultation of CEO. The list is sent to offshore team for ballpark estimate. The feature list is prioritized based on urgency and timelines set for release.

### Design

Based on the responsibilities assigned during the project initiation meeting, the project work is initiated. The principal Software architect prepares the system design specifications (SDS) document based on the approved requirements document. SDS document is the logical representation of the project requirements. Once SDS documents are prepared, then these are approved from the CEO to base line the requirements. SDS documents is sent to offshore team for review via e-mails and is also be shared via central repository.

Note: The SDS document is not developed for minor requirements and updates where the Work Plan/Proposal or User Requirement Specification document is adequate to provide high-level design specifications.

### Project Discovery

Once the project is initiated, the GM development assigns it to a Dev Lead for required project discovery activities. Discovery is an information-gathering process meant to dig deep into the details of what is important to a client's business, target audience, and industry. The Dev Lead starts this phase after project is approved. It involves putting together a team and assigning responsibilities to gather and develop all project requirement.

### High Level Project Plan

General Manager Development along with Dev Team Leads and QA Team reviews the SDS and the high-level project planning activities are carried out. These include developing the top-level work breakdown structure (WBS), estimating the project size, defining the project life cycle, and estimating the project cost. Following sections provide information on these activities:

#### Develop High Level WBS

The Development Lead, or the designated team member develops a high-level work breakdown structure (WBS) for the scope of work that needs to be completed for the given project. This WBS divides the overall project into an interconnected set of manageable modules or components. It also contains information on the tentative project timeline and high-level schedule. The WBS can be created in Microsoft Project or Microsoft Excel format. This WBS includes information on major deliverables related to discovery, design, development and project closure with their expected level of effort.

#### Estimate Effort and Cost

As part of this activity, project effort estimate is established using the initially created top-level WBS. This involves assigning estimated level of effort in terms of man hours to each sub-task and estimating the cost for major project milestones such as design, development, QA, deployment and project management. The level of effort for each sub-task is estimated based on the expert judgment. Project objectives with respect to effort estimate, are established using historic data and analytical techniques.

### Project Approval

Once the high level WBS is finalized, it is sent to Principal Software Architect and CEO for approval. Based on urgency, priority and time constraints, CEO may add or remove some of the work items in WBS. A project/release is started after it is approved from the CEO. The Principal Software architect prepares and sends the requirement specification document to offshore team for new features and/or where it is required.

The project approval may be received by several means which include

* email,
* telephonic conversation,
* On-shore and off-shore team meeting,
* comments/feedback on the shortlisted client requests.

After the WBS/project plan is approved by CEO, it is officially considered approved and project activities are formally initiated.

### Project Kickoff Meeting

After project approval, General Manager Development holds a kick-off meeting with key team members who need to be involved in design phase. These may include the technical team lead, graphic designers, and QA staff. Purpose of this meeting is to disseminate the initial project information, discuss high-level work plan among the team members, and assign responsibilities within the team. Meeting minutes of the Project Kickoff meeting are recorded.

### Detailed Project Plan

After the completion of the design documents and approval from client, detailed project plan is developed. Project is divided into phases which makes it easier to monitor. For this purpose, the development tasks are broken down to smaller workable items. Resources are allocated according to skillset required for each sub-task. The purpose of this document is to provide a comprehensive baseline of what has to be achieved and who is to be involved.

Following are the project management activities that are performed:

### Develop Detail Schedule

Based on the high-level WBS and estimates developed during high-level project planning, a detailed project schedule is developed. After completion and approval of the SDS documents from CEO, all implementation tasks are divided into smaller manageable development tasks. The detailed schedule includes sub-tasks to the deepest level possible. These tasks become the part of the detailed level WBS. As part of this activity, team resources are allocated and assigned for each sub-task. The detailed WBS also includes verification, validation, and release deployment activities.

### Plan Resources

The required project resources are planned. It includes hardware, software, and human resources. The knowledge and skills needed for the successful execution of the project are also identified. The type and extent of knowledge and skills to be provided to project staff is determined by assessing both available and needed knowledge/skillset. An appropriate option is selected from the following options:

• In-house training subject to availability of training expertise.

• External training.

• Staffing and new hiring.

Resources to be provided and the knowledge & skills to be acquired are measured periodically against the plan. Training schedule is incorporated into the project schedule if any such needs exist. Resource leveling and re-scheduling may take place based on resource training/skill requirements. The project resource planning involves following:

1. Determine process requirements.

2. Determine communication requirements.

3. Determine staffing requirements.

4. Determine facility, equipment, and component requirements.

5. Determine other continuing resource requirements.

### Plan Monitoring and Control

To keep the project on track, the project activities are monitored and controlled. Progress review meetings are planned by General Manager Development with all team members. In case there is any issue, desired corrective actions are planned accordingly.

#### Plan Progress Review

Project Manager plans the frequency and resources needed for project progress review meetings. The frequency of the progress review meetings is decided as per need base or as per the preference of on-shore and off-shore management. During progress review meetings, both project performance and project deviations are monitored, and corrective actions are planned.

#### Plan Stakeholder Involvement

Stakeholder’s commitments are identified and planned by project manager. These commitments are then monitored in the progress reviews meetings. Those commitments which are not satisfied are identified and results of commitment reviews are recorded.

#### Identify Project Risks

Risks may occur at any stage during the project life cycle. Risk Management Workbook is used to maintain risks and to identify impact/priority for involved risk. Risks are periodically reviewed throughout the project life cycle and are risk management workbook is updated when needed.

#### Plan Milestone Review

Project Manager plans the frequency and resources needed for milestone review meetings. The frequency of the milestone review meetings is decided as per need base or as per the preference of management or client. The accomplishments and results at selected project milestone are reviewed in milestone meetings and if there is any corrective action needed, it is planned and implemented as required.

### Project Execution

The project is executed based on the detailed project plan. The project execution incudes development, testing, and release activities.

#### Development

Actual development and implementation work is performed as per the detailed project plan. Every resource works according to the assigned tasks. During this phase, the development team provides feedback if any discrepancy is found in the project plan that requires to be updated.

#### Testing

After completion of the development and unit testing work from development team, internal QA is started. At this stage, the deployment packages for developed application along with database and other related components is provided to for QA for deployment on QA environment and testing and verification on staging environment. This is typically referred as alpha release. In house QCQA system is used to log the issues identified during internal testing. Verification and Validation plan is prepared and executed separately.

### Project Release Cycle

Following are the typical releases that are part of the project life cycle:

##### QA

The objective of alpha testing is to try and break the product, check the user requirements, and review the product in the light of the design specifications. Nexelus internal QA team performs the Alpha testing uses the product as though it was in production. The client can also be involved in early alpha releases depending upon the nature and volume of the project.

##### Release

After completion of the internal QA and relevant bug fixing by the development team, Release environments are prepared for QA to perform internal testing on environment identical to client. TFS is used to log the issues reported during the testing performed by client as part of release testing. In some cases, beta release memorandums are developed.

##### Test

The objective of Beta testing is for the customer and potential users to use the product with same data and environment as production system, in the real environment to determine that the system is ready for deployment and rollout. The objective of test Environment is for the customer to make the final acceptance decision based on the implementation of malfunction corrections as reported during the Beta phase. In some cases, acceptance release memorandums are developed.

##### Production

After completion of the acceptance release, Go Live activities are performed. In some cases, Go-Live memorandums and/or Deployment plans are developed. Please refer to Release Management Process for details.

### Conduct Monitoring and Control

To keep the project on track, the project activities are monitored and controlled. Development Lead conducts short meetings with development team for daily task status update. Periodic progress review meetings are also conducted with GM development to keep the project according to the plan. Peer reviews and code reviews are performed to ensure quality. Contingency plans are evaluated and implemented for any deviation from plan or risk observed. Milestone review meetings are also conducted with client to monitor the project progress and to get client feedback.

The project plan is updated based on the data collected during different project phases and progress review meetings. Following sections provided information on different activities performed for project monitoring and control:

#### Conduct Progress Review

Project progress and status is reviewed in periodic progress review meetings. All estimated tasks are evaluated, and progress is measured according to the project plan. Development Lead and QA Lead conduct this meeting with development and QA teams. Meeting minutes for each progress review meeting are recorded. Based on progress review meeting, corrective actions are taken and recorded to keep the project on track. The project progress review is performed on weekly basis between on-shore and off-shore teams. Project completion percentage is updated in project-plan, and it is updated as and when required.

#### Manage Stakeholder Involvement

Every project’s goal is the successful completion of the client’s requirements within specified timeframe. Management is kept involved by proving status updates on overall progress of the project during design, development, release and delivery tasks.

Development team conducts regular meetings to monitor the progress of the project and reviews any change that can affect the project plan. On-shore team is communicated the overall progress of the project. Any change suggested by the client or observed by the development team during development that can affect the project plan is communicated among all stakeholders.

#### Manage Project Risks

Project Risks include the internal risks and those risks that are beyond the control of project. Risk Management Workbook is used to maintain risks and to identify impact/priority for involved risk. Risk impact is calculated and prioritized using impact levels such as negligible, low, medium, and high levels. Risks are resolved according to the impact priority. High level risks are resolved first, and then medium and low-level risks are resolved by applying Contingency Plan.

#### Conduct Milestone Review

Milestones are used in project management to mark specific important points along a project timeline. These points may signal anchors such as a project start and end date, a need for external review or input and budget checks, among others. A milestone may be a single deliverable item or set of different items based on the requirements. Milestone review activity can be performed during design, development, release and delivery phases. PM and QA leads conduct this meeting with development and QA teams. The accomplishments and results at selected project milestone are reviewed in these meetings and if there is any corrective action needed, it is planned and implemented as required. Meeting minutes for each Milestone review meeting are recorded.

### Project Closure

The project closure phase is the last phase in project life cycle. After successful delivery of the project, the team reviews the project execution to see what went well and what could be improved. Project is analyzed and all the outcomes and lesson learnt are documented for future improvements. All the data collected from the project is added to the central repository for future reference.

# Release Management Process

This section explains process for deployment on client test and production environments

### Scope

Scope of this section is limited to deployment process on client test and production environments

## Roles and Responsibilities

### Release Manager

Release Manager is responsible for all deployments on client test and production environments. Other teams do not have access or rights to perform deployment on these environments

### QA Lead/QA Engineer

QA Lead/QA Engineer will assist Release Manager for process and feature testing if required.

### DB Lead/Database Developer

DB Lead/Developer will help troubleshoot any database deployment issues.

### Development Lead/Software Engineer

Development Lead/Engineer will assist release manager if he faces any web deployment issues.

## Process

Dot Net and DB will be responsible to place development packages in TFS under release management folder after complication of each project.

Every major / minor release will be placed on TFS under release Management folder which will be used to deploy release whenever needed, however for in-term releases / on-demand deployments during development phase, release will be built from current development methods.

|  |  |  |
| --- | --- | --- |
| Task | Description | Owner / Responsibility |
| Get latest Release package for DB and .Net | In-case of interim release, DB and .Net will provide package, otherwise package will be fetched from TFS. | Release Engineer |
| Development including DB scripts and .Net application along with supported services / application / configuration | Based on Time constraints 1 DB resource can be involved in DB deployments, i.e. for HealthStar where we need to upgrade more than 2 or 3 databases at a time. | Release Engineer |
| Customization scripts / Application changes. | DB scripts should be in TFS and  .Net resource can be utilized to apply application-level customizations | Release Engineer along with .Net resource |
| Verifying deployments by Smoke test | QA resource can be involved if more than 2-3 companies are being upgraded at a time. | Release Engineer |
| Testing if needed | 2 QA resources can be involved if there are more than 3 companies based on time constraints. | QA Resource |
| List of post Deployment Issues will be sent back to “Release Manager” |  | QA |
| Troubleshooting / Fixing Post deployment Issues | .Net and DB resource can be utilized to troubleshoot / Fix post deployment issues by release manager. | Release Engineer along with .Net and DB resource |
| Verification of post deployment fixes. |  | QA Resource |
| Any fix if there is any should be updated in DB / application package to make sure it won’t occur next time. | Release Manger should keep track of issues faced during deployment and should raise a hand if face same issue next time. | DB Manager / .Net Manager |
| Releasing application for offshore team to verify and or Hand over to client. |  | **Release Manager.** |
|  |  |  |

# Change Management

The process for managing application for change in system engineering is the process of requesting, determining accessibility, planning, implementation, and evaluating changes to the system. Its main objective is to support the processing and tracking of changes in a set of interconnected objects.

## Change Management Process

Nexelus implements 5-step change management process as describe in following chart.

Diagram

Description automatically generated

### Request for Change

The change request can be initiated by customer or Nexelus Management. The change request is sent to Principal Software Architect. The principal Software Architect then performs the following tasks:

1. Compiles change request list
2. Performs initial discovery and R&D
3. Prepares Requirement Specifications
4. Sends Requirement Specifications to CEO for Approval
5. After approval from CEO, Principal Software Architect creates Change Request Form and,
6. Sends it to GM Development for Impact Analysis

### Impact Analysis and System Design Specifications

Impact Analysis process is initiated once Requirement Specification and Change Request Form are sent to General Manager Development.

1. Based on Requirement Specifications, GM Development and Development Lead(s) perform Brain Storming session with CEO and Principal Software Engineer.
2. GM Development prepares impact Analysis and sends it for approval to CEO and/or, Client for Approval
3. Principal Software Architect presents the impact analysis to CEO for final approval. Once approved, the offshore team is given go-ahead for implementation
4. On Approval of Impact Analysis, GM Development and/or, Development Lead(s) prepare detailed System Design Specification and send it to Principal Software Engineer and CEO for approval
5. On approval of System Design Specification, GM Development prepares Work Breakdown Structure and estimate for change. If the change is over four weeks, then Milestone sheet is also created for US team to review progress.
6. Work Breakdown Structure and estimate is sent to CEO and/or Client for approval.

### Implement Change

Development cycle initiates after the Work Breakdown Structure and Estimate are approved.

1. GM Development assigns appropriate Team Lead and development team to implement change request.
2. Assigned Development Team implements the change,
3. If the estimate is over four weeks, then as per milestone worksheet, progress is reviewed by US Team.
4. In progress review meeting, some changes may be suggested by Review Team to be incorporated in System Specifications.
5. In case of substantial changes, GM Development updates the Work Breakdown Structure and, informs all stakeholders.
6. Once the change is implemented and unit tested, Development Lead creates the build and and hands over to QA Team for testing.
7. QA Team performs the testing and reports back to development team to fix if any issues are found in implementation.
8. If no issues are found, the QA team deploys the build on Release environment(s) and verifies the build. It reports back to Dev team if any issues are found.

### Review / Reporting

1. Once QA Team verifies that there are no outstanding issues in provided build on QA and Release environment(s), GM Development requests the Release Manager to deploy verified build on Test Environment.
2. GM Development requests Support team to inform client that change has been deployed on Test Environment for verification.
3. Client performs the verification on Test Environment and provides feedback.
4. On go-ahead from client, GM Development asks Release Manager to graduate the change on production environment.
5. Release Manager requests Support Department to inform all stakeholders, that change has been deployed on production Environment.

### Activity Diagram

