Novell Unified Delivery Process
Contents

1 Executive Summary ........................................................................................................................................ 1
2 Novell Unified Delivery Process Overview ................................................................................................. 3
3 The Dynamic: Iterations, Phases and Milestones ......................................................................................... 4
   3.1 Discovery Phase .................................................................................................................................. 5
   3.2 Inception Phase .................................................................................................................................. 5
   3.3 Elaboration Phase ................................................................................................................................ 6
   3.4 Construction Phase ............................................................................................................................ 7
   3.5 Transition Phase .................................................................................................................................. 8
   3.6 Iterations ............................................................................................................................................. 9
4 The Static: Activities, Workers, Deliverables and Tools .............................................................................. 10
5 Activity Threads ........................................................................................................................................... 12
   5.1 Scope .............................................................................................................................................. 13
   5.2 Direction Setting ................................................................................................................................. 14
   5.3 Architecture Design ............................................................................................................................ 15
   5.4 Requirements Assessment ............................................................................................................... 16
   5.5 Detailed Design ................................................................................................................................. 17
   5.6 Build, Configure and Develop ........................................................................................................... 18
   5.7 Integration Testing .............................................................................................................................. 19
   5.8 User Acceptance Testing ................................................................................................................... 20
   5.9 Deployment ....................................................................................................................................... 21
   5.10 Support .......................................................................................................................................... 22
   5.11 Supporting Activity Threads ............................................................................................................. 23
      5.11.1 Project Management .................................................................................................................. 23
      5.11.2 Knowledge Transfer ................................................................................................................ 24
6 Delivery Excellence Reviews ...................................................................................................................... 25
7 Summary ...................................................................................................................................................... 26
The Novell Unified Delivery Process (NUDP) is a solution delivery process based on the Rational Unified Process® (RUP) by IBM. While RUP is a software engineering process, NUDP takes the core principles and vocabulary from RUP and applies them to a broader solution space where the effort goes beyond pure application development.

NUDP is a disciplined approach to assigning tasks and responsibilities in the solution delivery effort. Its goal is to ensure high-quality solutions that meet the needs of Novell’s customers within a predictable schedule and budget.

The Novell Unified Delivery Process is a process definition developed and maintained by Novell Services. Novell Services continues to work closely with customers, partners and other groups within Novell to ensure that the process is continuously updated and improved to reflect recent experiences and proven best practices.

The Novell Unified Delivery Process enhances team productivity by giving every team member easy access to a consistent view of the solution delivery with the guidelines, templates, sample deliverables and tools for all critical development activities. By having all team members on the same process page, we ensure that all team members (whether working with requirements, designing, testing, or project management) share a common language, process, and view for delivering the solution.

The Novell Unified Delivery Process fits small project teams as well as large delivery structures. NUDP is founded on a simple and clear process flow that is common across a family of engagement types, yet it can be varied to accommodate different situations.

The Novell Unified Delivery Process is based on several solution delivery principles. These principles incorporate many of today’s solution-delivery best practices in a form that is suitable for Novell centric customer engagements of all sizes. Incorporating these best practices into the Novell Unified Delivery Process ensures that we are always doing the basic delivery items that have the biggest impact on our success. These best practices are commonly used by successful organizations. The Novell Unified Delivery Process provides each team member with the guidelines, templates and tools necessary for the entire team to take full advantage of the following best practices:

- Iterative solution development
- Manage requirements
- Use clearly defined architectures based on proven solution-reference architecture
- Verify solution quality
- Control solution changes

**Iterative solution development**

With today’s sophisticated environments and complex solutions, it is not possible to sequentially define the entire problem, design the entire solution, build the solution and then test the product in the end. The solution requires an iterative approach that allows an increasing understanding of the problem through successive refinements and an effective solution grown incrementally over multiple iterations. The Novell Unified Delivery Process supports an iterative approach to solution delivery that addresses the highest risk items at every stage in the lifecycle, significantly reducing a project’s risk profile. This iterative approach helps you attack risk through frequent, executable releases that enable continuous end-user involvement and feedback. Because each iteration ends with an executable release, the project team stays focused on producing results, and frequent status checks keep the project on track.
schedule. An iterative approach also makes it easier to accommodate tactical changes in requirements, features or schedule.

**Manage requirements**

The Novell Unified Delivery Process describes how to elicit, organize, and document required functionality and constraints, track and document trade-offs and decisions, and easily capture and communicate business requirements. The notions of use-case and scenarios proscribed in the process have proven to be an excellent way to capture functional requirements and to ensure that these drive the design, implementation and testing of the solution, making it more likely that the final solution fulfills the customer’s needs.

**Use clearly defined architectures based on proven solution reference architecture**

The process focuses on early development and base lining of a robust solution architecture prior to embarking on a full-scale solution construction. It describes how to design a resilient architecture that is flexible, intuitively understandable, accommodates change, and promotes a more effective solution. The Novell Unified Delivery Process uses proven reference architecture models as an input, ensuring that we always incorporate the most recent and up-to-date technology and process best practices in our solutions.

**Verify software quality**

Poor performance and poor reliability are common factors that make solutions unacceptable for today’s business users. Hence, quality should be reviewed for reliability, functionality, performance and availability. The Novell Unified Delivery Process assists you in planning, designing, implementing, executing, and evaluating these test types. Quality assessment is built into the process in all activities and with all participants. The assessment uses objective measurements and criteria and is not treated as an afterthought or a separate activity performed by a separate group.

**Control changes to the solution**

Managing change means making sure that each change is acceptable and being able to track changes in an environment where change is inevitable. The process describes how to control, track and monitor changes for successful iterative solution development.
2 Novell Unified Delivery Process Overview

The Novell Unified Delivery Process defines a flexible framework for delivering complex, business driven projects. The flexibility comes from the ability of the project manager to create a project plan matching the specific needs of the engagement while maintaining the disciplines required to align with delivery best practices.

It is the combination of dynamic and static constructs that allow the Novell Unified Delivery Process to be adapted to the specific needs of each engagement. The Novell Unified Delivery Process can be described in two dimensions:

- The horizontal axis represents time and shows the dynamic aspect of the process and is expressed in terms of iterations, phases, and milestones.
- The vertical axis represents the static aspect of the process: how it is described in terms of activities, workers, deliverables and tools.

Figure 1 shows both dimensions of the Novell Unified Delivery Process. The horizontal axis shows the phases of Discovery, Inception, Elaboration, Construction and Transition, along with their corresponding DE Review milestones. The vertical axis is represented by the activity threads of Scope, Direction Setting, Architecture Design, Requirements Assessment, Detailed Design, Build, Configure and Development, Integration Testing, User Acceptance Testing, Deployment, and Support.
3 The Dynamic: Iterations, Phases and Milestones

The Novell Unified Delivery Process divides the solution life-cycle into five phases.

- Discovery Phase
- Inception phase
- Elaboration phase
- Construction phase
- Transition phase

Each phase is concluded with a well-defined milestone—a point in time at which certain critical decisions must be made and therefore key goals must be reached. Each milestone is defined by a Delivery Excellence review where the goals and decisions of the phase are reviewed and acted upon.

![Diagram showing the five phases of the Novell Unified Delivery Process and the milestones occurring at the completion of each phase.](image)

Figure 2: Novell Unified Delivery Process Phases and Milestones

Figure 2 shows the five phases included in the Novell Unified Delivery Process along with the milestone occurring at the completion of each phase. There are two additional important points contained within Figure 2. First, most solutions will require multiple iterations of these phases, and the Novell Unified Delivery Process allows for this kind of flexibility. Second, you will notice the arrow showing a connection between the end of the Transition phase and the beginning of the Discovery phase. The arrow between the Transition and Discovery phases indicates that our process includes a consistent feedback loop allowing future iterations to benefit from the wisdom and knowledge gained by the previous.
3.1 Discovery Phase

During the Discovery phase, we establish the business needs for the solution and define the high-level project scope by identifying the customer’s business drivers and needs. The customer’s business drivers will give the context for structuring a high-level solution that defines boundary conditions and, most importantly, the scope for the engagement. The Discovery phase also includes the creation of initial project management artifacts (e.g. a project plan showing major milestones, risk assessment, and staffing model) based on the solution structure.

The deliverables of the Discovery phase are:
- A functional definition for the engagement constituting the solution definition, key features, main constraints and success criteria
- A business definition for the engagement constituting the financial plan (estimates for effort and pricing), initial risk assessment and a mitigation plan
- An initial project plan showing phases and iterations
- An internal Novell deal-review presentation
- A statement of work defining the contractual agreement between the customer and Novell for the engagement

Discovery Phase Delivery-Excellence Review

The evaluation criteria for the Discovery Phase Delivery-Excellence Review are:
- Customer stakeholder concurrence on the business definition, business value and customer resource investment.
- Customer stakeholder concurrence on scope definition and cost/schedule estimates
- Agreement on the project risks and mitigation plans
- “Go ahead” from customer stakeholders and Novell delivery management

The engagement may be canceled or considerably re-thought if it fails to pass this milestone.

3.2 Inception Phase

The Inception Phase is the phase where the business and IT requirements become clear, and a high-level design is created. This phase is crucial as it sets the foundation for the rest of the project. It includes creating detailed design documents, planning the project resources, and preparing for the execution of the solution. The deliverables of this phase would include a high-level design, project schedule, resource plan, and other planning documents.
The Inception phase is the delivery start of the engagement and includes launching the project and developing a deeper understanding of the customer's business environment and the architectural make up of the solution. The Inception phase will bring together a broader set of customer stakeholders and Novell participants than during the Discovery phase. This should include all parties playing a significant role in the engagement.

To accomplish this, we will conduct direction setting and architecture definition activities as required. The Inception phase includes kickoff activities to align the customer and Novell stakeholders on the project schedule and milestones, project assumptions, and customer responsibilities. The Inception phase also includes the refinement of the initial project management artifacts created during the Discovery phase.

The deliverables of the Inception phase are:

- Refined project management artifacts including a detailed project plan, risk assessment and mitigation plan, status reporting, and change control process definition
- Direction setting materials including current state assessment, future state recommendations, initiative prioritization, road mapping and business value justification, if needed
- Solution architecture design
- Completed project kickoff meeting

**Inception Phase Delivery-Excellence Review**

The evaluation criteria for the Inception Phase Delivery-Excellence Review are:

- Assess of the project risk and mitigation plans
- Assess of the customer’s readiness and availability of resources (both people and computing)
- Review the deliverables created during the direction setting and architecture definition activities
- Review any issues or to-do items based on the project kickoff activities
- Assess any changes to future phases (schedule, effort and/or cost)

### 3.3 Elaboration Phase

![Figure 5: Elaboration Phase](image)

The purpose of the Elaboration phase is to deeply analyze the problem domain, extend the high-level architecture definition and establish a sound, detailed design for the solution. It is during the Elaboration phase that we see the level of effort take a dramatic turn towards the detailed engineering decisions required to implement a successful solution.

At the end of this phase, the difficult "engineering" is considered complete. While the process must always accommodate changes, the Elaboration phase activities ensure that the architecture, requirements and plans are stable enough and the risks are sufficiently mitigated, so you can predictably move ahead with the completion of the solution.
In addition to requirements gathering and detailed design, the Elaboration phase activities include mapping business processes to the solution and documenting testing criteria. The Elaboration phase also includes a training and support assessment for the customer and the creation of a training plan for the customer.

The deliverables of the Elaboration phase are:
- A requirements-definition document including: solution use cases, business process maps, data flows, and technical requirements
- A detailed design document
- An updated architecture definition (as needed)
- Test criteria
- Updated project management artifacts (as needed)

**The Elaboration Phase Delivery-Excellence Review**

At this point, we will examine the detailed system objectives and scope, architecture and design choices, and the resolution of the major risks. The evaluation criteria for the Elaboration Phase Delivery Excellence Review are:
- Review solution scope and requirement priorities
- Review the detailed design for solution correctness and technical complexity
- Validate product placement and usage
- Review technical risks and validate their mitigation plans
- Review actual versus planned engagement expenditures
- Review and revise plans for next phase (as needed)
- Review support risks and mitigation plans

### 3.4 Construction Phase

During the Construction phase, we create the solution as defined by the deliverables from the Elaboration phase. The Construction phase is more of a manufacturing process where emphasis is placed on managing resources and controlling operations to optimize costs, schedules, and quality. In this sense, the management mindset undergoes a transition from the creative definition of the solution during inception and elaboration, to the deployment, configuration and construction of solutions during construction and transition.

Some projects are large enough that parallel construction phases can be spawned by the same Elaboration phase. These parallel activities can significantly decrease the time to deployable releases. The outcome of the construction phase is a fully implemented and tested solution ready for the end-users.

The deliverables of the Construction phase are:
- An implemented and fully tested version of the solution
- An Integration testing report
3.5 Transition Phase

The purpose of the Transition phase is to transition the solution to the customer. Once the solution has been given to the customer, Novell will assist the customer in completing user-acceptance testing and deployment into the target production environment. The Transition phase also includes the activation of support processes and resources that the customer will leverage after deployment. The Transition phase will also include any remaining training and knowledge transferring to customer resources.

The deliverables of the Transition phase are:

- User-acceptance test results and defect list
- A solution deployed into the target production environment
- Fully trained users, administrators, and operations staff
- Any agreed-to installation or operations documentation (agreed to in the SOW during the Discovery or Inception phases)

Transition Phase Delivery-Excellence Review

At this point, we will review the overall success of the project and plan future iterations and phases. The evaluation criteria for the Transition Phase Delivery-Excellence Review are:

- Assess user acceptance of the deployed solution
- Review operation characteristics of the solution in production
- Review actual versus planned engagement expenditures
- Obtain final sign-off of project deliverables
3.6 Iterations

Each engagement using the Novell Unified Delivery Process can be further broken down into iterations. An iteration is a complete solution implementation loop resulting in a release (internal or external) of the solution—a subset of the final solution under development—which grows incrementally from iteration to iteration to become the final system.

As mentioned earlier, most of the solutions implemented with NUDP span multiple iterations. It is just not pragmatic or reasonable to tackle an entire enterprise-class solution in one big “waterfall” process. These projects are always subject to delays, scope changes and other issues creating missed deadlines and over-budget efforts. By using iterations of the process shown here, we are able to deliver frequent, tangible results to our customers while keeping costs down and aligning with the core business drivers and business requirements.

Compared to the traditional waterfall process, the iterative process:

- Has clear objectives for each iteration. For example, "all architectural risks mitigated" or "all features required to support the HR on-boarding process built and tested". The iteration is not over until the objectives are met.
- Has a "just-in-time" mentality. An example would be to not produce the detailed requirements for a feature until you are ready to design, build, and test that feature.
- Understands that a deliverable is never "finished." It only "meets the objectives of the iteration." Solution requirements and design deliverables are living, breathing documents that will evolve with the solution in future iterations.
- Uses the risk and mitigation plan to decide what to do in early iterations. For example if the requirements for a feature are going to be difficult to pin down (they are identified as a risk), this feature should have its requirements detailed and be designed, built, and tested in an early iteration. Then the stakeholders can see it, and based on their feedback, a revision of the requirements.
- Can slightly overlap the iterations. Minimizes the whitespace for all resources by allowing them to focus on future iterations as they finish the current one.
- Has a dedicated project manager to perform the detailed iteration scheduling. The overall project plan is just a set of milestones and dates. The detailed planning of each iteration is not performed until the start of the iteration. However, this planning is driven from the risk and mitigation plan.
The Static: Activities, Workers, Deliverables and Tools

4 The Static: Activities, Workers, Deliverables and Tools

The static aspect of the Novell Unified Delivery Process defines the activity threads that contain the work in the phases and iterations. These activity threads contain the specific activities to be completed, the workers who will complete these activities, deliverables that will be produced as a result of the activities, and the tools the workers will use during the activity in the production of the deliverables.

This document will describe the activity threads in detail, but first; we need to describe the elements used as building blocks for each activity thread. There are four major elements within each activity thread:

- Activities: the “How”
- Workers: the “Who”
- Deliverables: the “What”
- Tools: the “Using”

Activities

An activity is a unit of work that an individual resource (see Worker) may be asked to perform. The activity has a clear purpose, usually expressed in terms of creating or updating one or more deliverables, such as a document, a product configuration, or a plan. Every activity is assigned to a specific worker. The granularity of an activity is generally a few hours to a few days. It usually involves one worker and affects one or only a small number of deliverables. An activity should be usable as an element of planning and progress; if it is too small, it will be neglected, and if it is too large, progress would have to be expressed by an activity’s parts.

Example of activities:

- Plan an iteration (Project Manager)
- Capture business requirements, for the Worker: Business Analyst, Strategist, Architect or Senior Technology Specialist
- Review the design, for the Worker: Architect or Principal Technology Specialist
- Execute performance test, for the Worker: Senior Technology Specialist or Project Manager

Workers

A worker defines the skills, experience and responsibilities of an individual working on an engagement. You could regard a worker as the role an individual can wear in the project. A worker performs a certain set of activities and owns a set of deliverables.

Example of workers typical of our engagements:

- Senior Project Manager, overall project lead and owner
- Senior or Principal Architect, project technical lead
- Senior Technical Specialist, Novell product specialists
- Senior Consultant, implementation resources
- Senior Strategist, Business planning and analysis specialists
Deliverables

A **deliverable** is a piece of information that is produced, modified, or somehow created by a process. Deliverables are the tangible products of the engagement; the things workers produce while working towards the final product. Deliverables are used as input by workers to perform an activity, and are the result or output of such activities.

Examples of typical deliverables:

- A **presentation slide deck**, such as a solution roadmap or solution architecture definition
- A **document**, such as a requirements definition or a detailed design document
- A **spreadsheet**, such as a business value justification, an initiative prioritization or a physical environment definition
- A **configured product**, such as the configured product or configuration scripts
- **Source code**
- **Executables**

Tools

A **tool** is a piece of information, technology or technique that is referenced or consumed by a process. Tools are tangible artifacts associated with a specific task on an engagement, the things the workers use while working towards creating the deliverables and final product(s) for the engagement.

Example of typical tools:

- **Deliverable Templates**, for maintaining consistency from engagement to engagement
- **Testing Harness**, for automating as much of the testing process as possible
- **Estimation Model**, for accurate, systematic and explainable estimates in complex solution engagements
- **Architecture and Design patterns**, for applying the right architecture and design constructs to the solution
- **Product Playbooks**, for applying the appropriate product implementation best practices
- **Product Management Tools**, for maintaining a consistent, high quality approach to project management from engagement to engagement

As we continue defining the static elements of the Novell Unified Delivery Process, it will be important to keep these definitions in mind. Each activity thread will be defined in the context of these four elements.
5 Activity Threads

The aggregate of all activities, workers, deliverables and tools does not quite constitute a working process. We need a way to describe meaningful sequences of activities that produce some valuable result, and we need a way to show interactions between workers.

We know that it is not always possible or practical to represent all of the dependencies between activities. Often two activities are more tightly interwoven than shown, especially when they involve the same worker, tools or deliverables. People are not machines, and the activity threads cannot be interpreted literally as a program for people to follow exactly and mechanically.

The Novell Unified Delivery Process contains ten primary activity threads and two supporting activity threads which represent all workers and activities partitioned into logical groupings.

![Figure 8: Primary Activity Threads](image)

Figure 8 shows both the primary activity threads and the supporting activity threads in a conceptual, sequential mode. We say this is conceptual because during an actual engagement some of the activity threads will overlap and some may repeat, depending on the circumstances.

The primary activity threads, as shown in Figure 8, are:

- Scope
- Direction Setting
- Architecture Design
- Requirements Assessment
- Detailed Design
- Build, Configure and Develop
- Integration Testing
- User Acceptance Testing
- Deployment
- Support

In addition to these primary activity threads, there are two supporting activity threads that continue throughout the life of an engagement. The supporting activity threads are:

- Project Management
- Knowledge Transfer

Together, these twelve activity threads provide the fabric for delivering solutions based on the Novell Unified Delivery Process. Although the names of the activity threads may remind us of the sequential phases in a traditional waterfall process, we should keep in mind that the phases of an iterative process are different and that these activity threads are revisited again and again throughout the life-cycle. The
actual complete flow of an engagement interweaves the ten activity threads and repeats them with various emphasis and intensity during each iteration.

**Primary Activity Threads**

The ten primary activity threads are defined in terms of the activities, workers, deliverables and tools associated with the thread.

### 5.1 Scope

The primary goal of the Scope thread is to come to agreement on the boundaries of the solution so that all parties involved (e.g. Novell, Customer, and/or Partner) can agree on what the solution will include and on the key assumptions. The activities in the Scope thread will include customer stakeholders to define and capture the solution objectives. We will define and capture the solution boundaries, including organizational areas, users and systems, and determine and communicate the Novell products required for the engagement.

**Activities**
- Review business statements and dimensions
- Determine involved systems, processes and people/identities
- Define delivered capabilities
- Review budget
- Create plan to address dependencies, risks, communication, and sponsors

**Workers**
- Services Principal
- Project Manager
- Architect
- Strategist
- Practice Lead

**Deliverables**
- Proposal deck
- Deal review template (DRT)
- Objectives / intro section in SOW
- Free-standing statement of scope / project charter

**Tools**
- Customer interviews
- Statement of work and pricing tool templates / Estimation worksheets
5.2 Direction Setting

The Direction Setting thread is focused on establishing a clear, business-driven direction for the customer within the technology area targeted by the engagement. Novell will work with the customer’s business leaders to create a consensus on business and technical initiatives that will benefit from using Novell technology as part of a solution. In order to create this consensus, Novell will define a conceptual future state architecture along with an implementation road map showing the customer the major activities required to move to the recommended future state. If needed, Novell will also create a financial justification allowing the project to receive the necessary customer approvals. All throughout this phase, Novell will continue educating the customer on the business value and technical merit of the Novell-centric solution.

Activities

• Validate business drivers
• Validate solution goals & objectives
• Assess current state technology, processes and skills
• Develop future state recommendations
• Perform a current state architecture assessment
• Define a conceptual future state solution architecture
• Identify and prioritize initiatives
• Create business value justification
• Create gap analysis (current state vs future state)
• Develop a recommended solution road map
• Develop a resource plan

Workers

• Architect
• Strategist
• Technical Specialist (as needed)

Deliverables

• Initiative portfolio and prioritization
• Solution area road map and resource plan
• Future state solution architecture
• Business value justification assessment

Tools

• Reviewing customer technical, business, organization, and audit documentation
• Customer Interviews
• Business and technology questionnaire response(s)
5.3 Architecture Design

The Architecture Definition phase creates a detailed technical foundation for the solution. This process includes the definition of the solution architecture based on the business context gathered during the Direction Setting and Scope threads. Additional requirements (both business and technical) will be gathered as necessary during the architecture definition thread. The architecture definition consists of a number of architectural views. These views capture the major structural design decisions. At their core, architectural views are abstractions or simplifications of the entire design, in which important characteristics are made more visible by leaving details to a later time.

The architecture is an important vehicle for both creating an accurate, detailed design and increasing the quality of the solution build. The architecture definition thread will also communicate the gaps between the current state architecture and the future state architecture.

Activities

- Review current architecture and existing technologies within the solution area
- Review current IT strategy and planned technical initiatives
- Review business strategy for future state architecture impact (e.g. acquisitions, divestitures, etc.)
- Conduct interviews with key stakeholders who have both technical and business responsibilities
- Define future state architecture
- Assess gaps between the current state and the defined future state architecture

Workers

- Project Manager
- Architect
- Architect
- Technology Specialist (as needed)

Deliverables

- Future state architecture definition including:
  - Conceptual architecture
  - Data model / architecture
  - Logical architecture
  - Physical/security architecture
  - Deployment architecture
  - Gap analysis
5.4 Requirements Assessment

The goal of the Requirements Assessment thread is to describe what the solution should do and allow the team members and the customer to agree on that description. To achieve this, we capture, structure, and document the necessary functions and constraints. We document the trade-offs and decisions that define what will specifically be in the solution. The requirements gathered during this phase are the foundation for how the solution is constructed and also how Novell should support the solution after implementation. To this end, Novell will also seek to understand the customer’s organizational structure, support/operational processes, and behavior.

During this phase, Novell will define the use cases, business constraints and technical requirements for the solution. Novell will define acceptance criteria to be used as the basis for design and automated testing later in the engagement life-cycle.

Activities

- Validate solution goals & objectives
- Conduct customer interviews and review available customer documentation
- Define resource requirements (hardware, software, and people)
- Create the requirements assessment report
- Prioritize requirements (in relative order)
- Create acceptance criteria documents
- Map business processes to the solution
- Assess customer support and training requirements

Workers

- Project Manager
- Architect
- Strategist
- Technical Specialist

Deliverables

- Requirements assessment report
- Business process maps
5.5 Detailed Design

The goal of the Detailed Design thread is to show how the system will be realized during the following phases. We want to make sure we build a system that meets all the defined requirements and performs the tasks and functions laid out in the specified production environment. The detailed design for the solution is architecture-centric, with a definition based on the architectural blueprint created in an earlier thread. The production and validation of the architecture is a prime focus of the Detailed Design thread.

It is during detailed design that we validate the hardware and software specifications for all labs and production environments. We also start to execute the training plan created in the requirements assessment phase.

Activities
- Conduct customer interviews and review available customer documentation
- Validate and update resource requirements (hardware, software, people) for all labs and production environments
- Create the detailed design document
- Conduct customer training classes (as needed)

Workers
- Project Manager
- Architect
- Technical Specialist (as needed)

Deliverables
- Solution architecture update (as needed)
- Detailed technical design for the solution
- Training materials (as needed)

Tools
- Customer interviews
- Solution Architecture definition
5.6 Build, Configure and Develop

The name of this thread, Build, Configure and Develop, specially describes the purpose of the thread. A more general software development lifecycle definition, such as RUP, would refer to this thread as implementation or development, but because of the Novell product-centric nature of the Novell Unified Delivery Methodology, we use a thread name that is more descriptive of the activities contained within.

- **Build the solution environments** required to deliver the solution as defined by the previous threads.
- **Configure the products** (Novell, open source, and third party) required to deliver the solution as defined by the previous threads.
- **Develop any custom code or scripts** required to deliver the solution defined by the previous threads.

In addition to the activities aligned under the Build, Configure and Develop terminology, this phase includes a robust unit-testing process.

**Activities**

- Build configuration environment
- Load/configure an appropriate automated testing harness
- Install and configure solution functionality in development and test labs
- Perform any development activities (as needed)
- Perform unit tests
- Validate test results
- Create code fixes and retest
- Build deployment and/or migration processes, test data and scripts
- Write integration test cases
- Support customer in defining user-acceptance test cases

**Workers**

- Project Manager
- Architect
- Technical Specialist (as needed)

**Deliverables**

- Integration and user acceptance test cases
- Installation and configuration of lab environments
- Unit-tested solution components
• Known defect report
• Deployment plan, processes and test data

Tools
• Engagement scope definition (to establishing boundary conditions for building test cases)
• Solution architecture definition
• Detailed design document
• Production, testing and development lab specifications
• Acceptance criteria documents

5.7 Integration Testing

The Integration Testing thread ensures that all the individually developed and unit-tested solution components work well together, containing the necessary functions and meeting the technical demands of the solution.

Integration Testing:
• Verifies proper software component integration
• Verifies that all requirements have been correctly implemented
• Identifies and ensure defects are addressed prior to the deployment of the solution

The Novell Unified Delivery Process dictates an iterative approach, which means that you test throughout the solution creation lifecycle. This allows you to find defects as early as possible, which radically reduces the cost of fixing the defect. Tests are carried out along three quality dimensions: reliability, function, and performance. For each of these quality dimensions, the process prescribes that we go through the test lifecycle of planning, design, implementation, execution and evaluation.

Novell’s approach to testing also includes the use of test automation wherever possible. Automated testing is especially important when using an iterative approach because it allows regression testing at the end of each iteration and with each new version of the solution.

Activities
• Load and/or configure test harness (as needed)
• Update and test lab environment (as needed)
• Perform integration tests
• Validate test results
• Update solution and retest
• Create known defect list
• Update deployment and migration processes, test data and scripts (as needed)
• Create integration testing results report

Workers
Activity Threads

- Project Manager
- Technical Specialists (as needed)

Deliverables
- Integration tested solution components
- Known defect list
- Integration testing report

Tools
- Integration test cases
- Acceptance criteria documents
- Data cleanliness evaluation report
- Test harness (as needed)
- Integration test results dashboard

5.8 User Acceptance Testing

User acceptance testing begins the transition of a specific iteration to the customer’s ownership. The user acceptance thread allows the customer to validate that the pre-defined business scenarios function properly within the solution. This provides a context for the customer to agree (sign-off) that all the required functions are being present and working in the solution.

Novell will transfer knowledge to the customer resources performing the user acceptance testing. This allows the customer to identify and fix any final defects in the solution prior to deployment.

Activities
- Update test labs (as needed)
- Perform user acceptance tests
- Validate test results
- Update solution and re-test
- Update known defect list
- Update deployment and migration processes, test data & scripts (as needed)
- Create user acceptance testing results report

Workers
- Project Manager
- Architect
- Strategist
- Technical Specialist
- Practice Lead
5.9 Deployment

The purpose of the Deployment activity thread is to produce a successful release of the solution and deliver the solution to its end users. This includes installing the solution into the customer’s production environment, performing all necessary migration and validation tasks, and providing knowledge transfer to the customer’s end users and operation teams.

Although deployment activities are mostly centered around the Transition phase, many of the activities are included in earlier phases to prepare for deployment when the project plan dictates.

Activities

• Promote the solution functionality in the customer’s target environment according to the go-live process
• deploy go-live
• Execute the production migration plan
• Monitor and resolve any post-production issues
• Perform final knowledge transfer to customer

Workers

• Project Manager
• Architect
• Technical Specialists
• Deliverables
• Production solution deployment
• Final solution knowledge transfer
• Production deployment documentation

Tools

• Successful integration and UAT test results
• Solution rollback plan

Figure 17: Deployment
5.10 Support

The Support activity thread is different than the other activity threads. The support thread does not have a specific end-date associated with the completion of the engagement but continues on for a length of time determined by the customer's support contract with Novell. It is common for support to be ongoing as additional iterations or brand new engagements are delivered to the customer. The goal of the Support activity thread is to ensure that the solution meets the customer’s up-time SLA and availability demands, and to limit and resolve product issues as they arise. We will also continue to educate the customer about the solution and how they can best use Novell technology to resolve business problems as they arise. The workers delivering the Support activity thread often become trusted advisors to the customer.

Activities
- Support the deployed solution to the customer's purchased support level
- Resolve and limit product Issues
- Transition product defects and enhancements identified during the project lifecycle
- Pro-actively maintain the health of the solution

Workers
- Field Support Engineer
- Services executive

Deliverables
- Support activity reports (weekly, monthly, etc.)
- Product issue reports
- Customer satisfaction results (surveys, interviews, etc.)
- Additional opportunity definition (e.g. sew solutions, additional phases on existing solution, etc.)

Tools
- Known defect list
- Debugging and troubleshooting tools
- Novell solution best practices knowledge base
5.11 Supporting Activity Threads

5.11.1 Project Management

Project management is the art of balancing competing objectives, managing risk and overcoming constraints to successfully deliver a solution that meets the needs of both customers (solution owners and administrators) and the end users who use the solution to complete their daily work assignments. This is not a task to be taken lightly. The fact that so few projects are unarguably successful is comment enough on the difficulty of the task.

This activity thread cuts across all of the other activity threads and phases to provide a consistent, agile, flexible approach to managing complex solutions.

Activities

• Create and manage project schedules
• Create and manage risk assessment and mitigation plans
• Create quality management approach (Identify the types of project reviews and walkthroughs that will be conducted, both internally and externally)
• Change management
• As needed, oversee and direct customer tasks that affect project completion
• Create, baseline, and manage project financials
• Produce regular project-status communication
• Lead communication with the customer and Novell on the overall status of the project and the resolution of specific project issues
• Project Closure (project evaluation, sign-off, team reviews, archive)

Workers

• Project Manager
• Practice Lead

Deliverables

• Accurate, updated project plan(s)
• Risk and mitigation plans
• Communication Plans
• Known issue list and tracking sheet
• Engagement status reports

Tools

• Project management software
• Project management templates and guidelines
Novell believes that educating our customers during the course of our engagements is a critical part of a successful project and our long-term relationships with our customers. This activity thread is shown as a supporting activity thread, covering the full lifecycle of the methodology because knowledge transfer activities are included in almost every primary activity thread. Knowledge transfer continues during the Support activity thread for as long as Novell is engaged with the customer.

Activities
- Mentor customer resources on Novell technology
- Communicate Novell solution best practices
- Communicate Novell product enhancements

Workers
- Field Support Engineer
- Services executive (SAM)
- Architect
- Project Manager
- Technical Specialist
- Strategist

Deliverables
- Verbal mentoring of customer personnel
- Brown bag learning sessions
- Documented best practices

Tools
- Novell solution best practices knowledgebase
- Novell product roadmaps
6 Delivery Excellence Reviews

Novell uses the Delivery Excellence process as a means of maintaining quality and consistency across our solutions. The Delivery Excellence reviews are comprehensive reviews, each occurring after each phase in the project lifecycle. The reviews include resources from both Novell services and Novell engineering; a common, accepted view of solution delivery and support.

The length and method (e.g. face-to-face, conference call, etc.) of the Delivery Excellence review will vary based on the project complexity and the phase being reviewed, but all projects will adhere to the review cadence defined in the Novell Unified Delivery Process.

As indicated in Figure 21, Delivery Excellence reviews occur after each phase in the Novell Unified Delivery Methodology. Each Delivery Excellence review has a slightly different focus and different deliverables. These are spelled out in detail in the project phase sections of this paper. The Delivery Excellence reviews are conducted by a cross section of Novell resources, including:

- The full Novell services team involved with the customer engagement. This includes resources from consulting, support (DSE/PSE), and partners who are involved in the project.
- Peer review from a full Novell services team including representatives of the Novell services architecture, project leadership and business-unit solution practices.
- Novell engineering resources from the appropriate business unit(s)
- Field sales representatives that are active with the engagement or account (as needed)

The results of all Delivery Excellence reviews are aggregated into a set of action items for the project manager to track and address. This review is available to all Novell, partner and customer management.
The Novell Unified Delivery Process is a process created and executed by Novell to ensure successful delivery of client engagements using Novell products and associated technologies. We discussed how the strength and flexibility of the Novell Unified Development Process is due to the two distinct axis of defined activities:

- The dynamic axis is composed of the engagement phases. These phases can be structured in different ways depending on the demands of the customer and the engagement. These phases are organized into increments, allowing the working parts of the solution to be delivered as frequently as possible. The phases are:
  - Discovery
  - Inception
  - Elaboration
  - Construction
  - Transition

- The static axis is composed of the primary and support activity threads. These activity threads provide the detailed activities that must occur within each phase for the project to be successful. The activity threads are:
  - Scope
  - Direction Setting
  - Architecture Design
  - Requirements Assessment
  - Detailed Design
  - Build Configure and Develop
  - Integration Testing
  - User Acceptance Testing
  - Deployment
  - Support
  - Project Management
  - Knowledge Transfer

The Novell Unified Delivery Process allows for priority driven, goal-focused, and change-control managed engagements. It allows for proof of concepts, pilots (production or tests), prototypes, alpha and beta releases, and full production releases. This process, combined with architectural, design and product configuration best practices have been developed by Novell to limit risk and ensure the success of engagements for our customers and with our partners.