ServiceNow White Paper: Migrating from Cherwell Service Management

Executive Summary

Moving from Cherwell Service Management to ServiceNow is a big step, but it's one that can bring a ton of benefits. ServiceNow offers a modern, cloud-based IT Service Management (ITSM) platform with built-in automation, AI capabilities, and best-practice workflows. This guide will walk you through common migration challenges, a pre-migration checklist, key roles for planning, and a baseline project plan. We'll also discuss how customizations in Cherwell can affect the complexity of the migration and how to transition your existing resources smoothly. Plus, we'll highlight what you can expect when it comes to User Interface (UI) and User Experience (UX) changes and how to keep your team engaged with Organizational Change Management (OCM).

Common Challenges in Cherwell to ServiceNow Migration

Switching from Cherwell to ServiceNow isn't always straightforward. Here are some common challenges you might run into:

- **Data Mapping & Integrity**: Making sure your data translates correctly from Cherwell's structure into ServiceNow without losing important details.
- **Customization Complexity**: If your Cherwell setup is heavily customized, you may need to rethink or rebuild workflows in ServiceNow.
- **Process Alignment**: Some processes may work differently in ServiceNow, requiring adjustments to workflows.
- **Integration Dependencies**: Any third-party integrations will need to be reviewed, redesigned, and tested in ServiceNow.
- **User Adoption**: Users will need time to get used to the new platform, so training and support are crucial.
- **Performance Optimization**: Migrated data and workflows need to be optimized for ServiceNow's cloud environment.

Migrating Cherwell Customizations vs. Adopting ServiceNow Out-of-the-Box Features

One of the biggest decisions during migration is whether to migrate existing Cherwell customizations or adopt ServiceNow's out-of-the-box (OOB) capabilities. So, should you haul all your old stuff over or embrace the shiny new features of ServiceNow? Let's break it down.

Migrating Cherwell Customizations

Pros:

Keeps things familiar – Your team won't have to relearn everything from scratch, which means fewer complaints in your inbox.

Preserves business-specific processes – If your organization has spent years perfecting workflows, this avoids reinventing the wheel.

Maintains legacy integrations – Some integrations are like that one office printer that never dies—still critical, still working.

Cons:

Expensive and complex – Custom development means extra costs, extra work, and extra time.

Performance risks – Not everything built for Cherwell will play nicely in ServiceNow's cloud-based environment.

Slower implementation – Testing and debugging custom scripts can take ages, delaying the go-live date.

Adopting ServiceNow Out-of-the-Box Features

Pros:

Best practices baked in – ServiceNow's workflows are designed for efficiency, so why not use them?

Faster implementation – No custom coding means you can get up and running much quicker.

Easier upgrades – You won't have to worry about breaking your customizations every time there's a new release.

More stable and secure – Using what's already there means fewer system hiccups and compliance headaches.

Cons:

Change is hard – Users might need some time (and patience) to adjust to new processes.

Might not cover everything – Some unique business requirements may still need customization.

- Reengineering required – Your old ways might not map perfectly to ServiceNow's workflows.

Methodology for Evaluating Customization Migration

So, how do you decide whether to migrate customizations or go all-in with ServiceNow's OOB features? Here's a handy framework:

1. **Assess Business Criticality**
    - Is this customization something your business absolutely needs, or is it a "we've always done it this way" situation?
    - Does it give you a competitive advantage, or could you work around it?
    - Would a standard ITSM workflow work just as well?

2. **Evaluate ServiceNow Functional Equivalents**
   - Check if ServiceNow already has a built-in feature that does the job.
   - Consider minor tweaks to OOB features instead of full customization.
3. **Analyze Maintenance & Upgrade Costs**
   - Custom solutions need ongoing support, while OOB features get automatic updates from ServiceNow.
   - Long-term, is maintaining a custom workflow worth the effort?
4. **Determine Performance & Scalability Impact**
   - ServiceNow is optimized for OOB functionality—custom solutions might not run as efficiently.
   - Will your customizations slow things down or create unnecessary complexity?
5. **Test User Adoption & Change Management Needs**
   - How much training and support will be required for a process change?
   - Pilot test both approaches and see which one users prefer.

Following this structured approach will help you figure out what's worth keeping and what's better left behind. Remember, sometimes it's better to let go of the past and embrace the future—especially when the future means fewer headaches and smoother upgrades.

Pre-Migration Checklist

To set yourself up for a smooth migration, follow this checklist:

1. **Assess Your Cherwell Environment**
    - Identify all configurations, customizations, and workflows.
    - Determine which data needs to be migrated.
    - **Comprehensive List of Data Elements & Customizations to Evaluate:**
        - **Core Data:** Incident records, change requests, service requests, assets, CMDB, knowledge base articles, user and group information.
        - **Configurations:** Business rules, security roles, form layouts, email notifications, approval processes.
        - **Customizations:** Custom fields, dashboards, reports, scripted automations, custom actions.
        - **Workflows & Automation:** Custom business processes, event-driven actions, integrations with external applications.
    - **Methodology to Categorize Complexity:**
        - **Low Complexity:** Out-of-the-box configurations with minor modifications (e.g., simple form layout changes, standard field mappings).
        - **Medium Complexity:** Some level of scripting and automation required, moderate custom workflows, integrations with common third-party tools.
        - **High Complexity:** Extensive scripting, custom-built applications, heavily modified workflows, deep integration dependencies.
2. **Define Your Goals**
    - What are the key business outcomes you want to achieve?
    - Which ServiceNow modules will you be using? (Consider **ITSM, ITOM, ITAM, CSM, HRSD**.)
3. **Assemble the Right Team**
    - Engage executive sponsors, IT leaders, and process owners.
    - Assign technical resources like ServiceNow admins and developers.
4. **Understand Customization Impact**
    - List out Cherwell scripts, automation, and integrations.
    - Determine which can be replaced with out-of-the-box ServiceNow functionality.

5. **Develop a Data Migration Strategy**
    - Plan how data will be extracted, transformed, and loaded into ServiceNow.
    - Clean up old or unnecessary records.
6. **Plan for Integrations**
    - Identify third-party integrations and plan for their rebuild in ServiceNow.
7. **Prepare Your Users**
    - Develop an Organizational Change Management (OCM) strategy.
    - Start training early and communicate changes regularly.

Subject Matter Experts (SMEs) and Resources Required

Subject Matter Experts (SMEs):

- **ServiceNow Architect** – Designs overall system architecture and best practices.
- **Cherwell Administrator** – Provides expertise on existing Cherwell configurations.
- **Data Migration Specialist** – Ensures smooth data extraction, transformation, and loading.
- **Business Process Owner** – Ensures process alignment with business objectives.
- **Integration Engineer** – Manages third-party integrations and API connections.
- **Security & Compliance Expert** – Oversees SOX, SOC, and data integrity requirements.
- **Training & Change Management Specialist** – Develops and implements user training and adoption strategies.
- **Project Manager** – Oversees migration timelines, risks, and coordination.

Resources Required:

- **ServiceNow Development & Test Environments** – For configuration, testing, and validation.
- **Cherwell System Documentation** – Helps understand existing workflows and configurations.
- **Data Migration Tools** – Ensures structured and validated data migration.
- **Integration Platforms (e.g., Middleware, APIs)** – Supports system connectivity.
- **Training Materials & Learning Management System (LMS)** – Facilitates end-user training.
- **Change Management & Communication Tools** – Supports engagement and smooth transition.
- **Testing & UAT Environment** – Ensures comprehensive validation before go-live.

Key UI/UX Differences

Migrating from Cherwell Service Management to ServiceNow isn't just a tech upgrade—it's a big shift in how users interact with the system. The UI and UX differences between the two platforms can be pretty dramatic, but understanding these changes ahead of time can make the transition smoother.

This guide breaks down the key UI/UX differences, including navigation, customization, workflows, and reporting, with real-world examples to help put things into perspective.

1. Navigation & User Interface

Cherwell:

- Feels more like an old-school Windows application with a desktop-style layout.
- Users navigate through a tree-style menu, which can get overwhelming as more options are added.
- You often need to open multiple windows to complete a single task—think of managing an incident while pulling up related assets in separate pop-ups.

ServiceNow:

- Has a cleaner, web-based UI with a left-hand navigation bar that keeps things organized.
- Features a global search bar that lets users find records and actions quickly, instead of clicking through menus.
- Tasks like creating incidents or updating change requests happen in a single interface, reducing clutter and making multitasking easier.

Example:

In Cherwell, if you need to reassign a ticket, you might have to go through multiple dropdowns and confirm changes in a pop-up window. In ServiceNow, you just type the assignee's name in the designated field and save—it's that simple.

---

2. Customization & Design

Cherwell:

- Allows deep customization, but it requires a lot of manual setup and scripting.
- The form editor can be tricky—fields and buttons don't always align perfectly, and changes can affect multiple workflows unintentionally.
- Mobile support is minimal, often needing extra configurations to work smoothly on smaller screens.

ServiceNow:

- Uses a modern, mobile-friendly design with UI policies that make customization much easier.
- Admins can make quick field and form changes using a drag-and-drop editor without worrying about breaking existing workflows.
- Fully responsive, so whether you're on a desktop, tablet, or phone, everything adjusts accordingly.

Example:

In Cherwell, if you want to create a custom incident form with conditional fields (e.g., showing a "Root Cause" field only if "Resolved" is selected), you'd have to set up multiple rules manually. In ServiceNow, you can do this with a simple UI policy, no scripting needed.

3. Workflow & Process Automation

Cherwell:

- Uses One-Step Actions to automate tasks, which work well but require a lot of setup.
- Workflows can be built using a visual editor, but complex processes often involve nested conditions that are hard to track.
- Many automations rely on batch processing, meaning updates don't always happen in real time.

ServiceNow:

- Uses Flow Designer, a more intuitive drag-and-drop automation tool that's easier to maintain.
- Workflows can trigger real-time updates, sending alerts or making assignments instantly.
- AI-driven features, like Predictive Intelligence, help automate ticket categorization without manual setup.

Example:

In Cherwell, if you need to auto-assign tickets based on category, you might have to create multiple One-Step Actions and business rules. In ServiceNow, you can set up an Assignment Rule that does this dynamically based on category, priority, and other factors.

4. Reporting & Dashboards

Cherwell:

- Uses a traditional report builder with static dashboards.
- Custom reports often require SQL knowledge, making them tricky for non-technical users.
- Limited visualization options—mostly tables and basic charts.

ServiceNow:

- Offers real-time dashboards with interactive widgets that update instantly.

- Drag-and-drop report building makes it easy for anyone to create reports.
- Performance Analytics provides deeper insights, allowing for trend tracking and predictive insights.

Example:

If you need to track SLA compliance, Cherwell might require a custom-built SQL query to generate a report. In ServiceNow, you can simply select the SLA table, apply filters, and generate a visual dashboard in minutes.

5. Mobile & Accessibility

Cherwell:

- Mobile experience is pretty basic—limited functionality, and some forms don't scale well on smaller screens.
- No dedicated mobile app, making it harder to use on the go.

ServiceNow:

- Fully mobile-optimized with a dedicated app that lets users manage incidents, approvals, and requests from anywhere.
- Provides push notifications, ensuring users stay updated without needing to check emails constantly.

Example:

A field technician using Cherwell might struggle to update an incident from their phone due to formatting issues. With ServiceNow's mobile app, they can update ticket status, add comments, or upload pictures directly from their device.

Conclusion

Switching from Cherwell to ServiceNow isn't just about getting a new system—it's about making IT service management smoother, more efficient, and user-friendly. With its modern UI, real-time automation, and easy customization, ServiceNow makes life easier for both IT teams and end users. Understanding these key differences will help teams get up to speed faster and make the most of their new platform.

Reporting Considerations

Migrating reports from Cherwell to ServiceNow sounds like fun, right? Kidding—it's definitely a bit of a process. But don't worry, we've got you covered. It's not just about copying reports; it's about making sure your data stays intact, your users don't revolt, and your team gets the insights they need.

Let's break it down into bite-sized chunks (because who has time for more?).

---

1. First Things First: Know the Difference

Before you dive headfirst into the migration madness, it's good to know the difference between how Cherwell and ServiceNow handle reports. It's like switching from a flip phone to a smartphone—there are a lot more features, but it's not exactly the same.

- Cherwell Reporting: You've probably spent hours with Crystal Reports and running SQL queries. Cherwell's report builder isn't *bad*, but it's pretty old school.
- ServiceNow Reporting: Welcome to the future! ServiceNow uses its built-in Report Designer, and for those flashy, real-time, dashboard-style reports, there's Performance Analytics (PA). This is where things get fun—lots of graphs, trends, and fancy visuals.

So yeah, it's not just a copy-paste job. You've gotta adjust.

---

2. The Actual Migration Process: Let's Get Down to It

Okay, here's how you migrate those Cherwell reports to ServiceNow. Spoiler: It's not like transferring your playlists from Spotify. There's some work involved.

A. Catalog Your Cherwell Reports

- What to Do: List every single report that exists in Cherwell. You'll want to know which ones are used most often and which ones are, well, kind of pointless. That's where you can start trimming the fat.
- Example: You've probably got a "Service Desk Performance" report in Cherwell that tells you how many incidents were resolved in the last month. Is that going to move to ServiceNow? Probably. But are you going to keep the one that tracks incidents closed in the last 30 minutes of every workday? Let's be real—probably not.
- Tip: Ask your users—they'll know which reports are actually useful, and which ones they only look at once a year (and even then, they don't really read them).

B. Understand ServiceNow's Data Architecture

- What to Do: ServiceNow works with Data Sources to pull data into reports. These are kind of like the engine of your reports—if your data isn't flowing properly into the engine, good luck with your report.

- Example: Let's say you've got a report in Cherwell that pulls incident data. In ServiceNow, you'll have to set up a Data Source that pulls from the Incident table. If the data is coming from some other system (like an external tool), you'll need to set up integrations or REST APIs.
- Tip: If you're working with a massive set of data, you might need to use Performance Analytics to track things like incident trends, because simple reporting could get messy real quick.

C. Recreate or Redesign Reports

- What to Do: Now that you've mapped out your data, it's time to get down to report creation. Simple reports are easy to redo in ServiceNow's Report Designer. If you're used to doing fancy SQL queries, you'll have to translate that into ServiceNow's system.
- Example: You had a super complex "Incident Resolution Time by Severity" report in Cherwell that pulled data from multiple tables with SQL. In ServiceNow, you'll likely need to rebuild that with a Data Source and potentially a Performance Analytics widget for more interactive data exploration.
- Tip: Don't forget to tweak the formatting. ServiceNow isn't as flexible as Cherwell with custom report layouts (no more Crystal Reports, sorry!), so you'll want to make sure things look good and are easy to understand for the end users.

D. Test, Test, and Test Again

- What to Do: Once you've created your reports, test them out. Make sure the numbers add up, and be ready for a few "wait, that's not right" moments.
- Example: You recreated that "SLA Compliance" report, but the numbers look off. Maybe you missed a filter or didn't account for one tiny little variable in the data. Test with real-world scenarios and get feedback from the folks who actually use the reports every day.
- Tip: User feedback is everything. If your users aren't happy with the results, they're not going to use the new reports.

---

3. Common Issues to Watch Out For (And Fix Before You Panic)

Migrating reports is never a straight line. Expect some bumps along the way. Here are a few things that could trip you up:

A. Data Mismatches

- Problem: ServiceNow and Cherwell store data differently. For example, one system might treat incident priority as a numeric value, while the other might treat it as a drop-down with text options. This can cause discrepancies in your reports.
- Fix: Plan for data transformations. You'll likely need to adjust how data is pulled and presented, so keep that in mind early on.

B. Formatting Woes

- Problem: Cherwell's reporting is more flexible with formats—especially Crystal Reports. ServiceNow is powerful but doesn't give you the same level of customization.
- Fix: Don't try to replicate every little design feature. Instead, focus on delivering clear, insightful data that's easy for users to digest.

C. Performance Problems

- Problem: ServiceNow is awesome, but if you try to pull a mountain of data at once, you might experience some lag or performance issues.
- Fix: Use Data Sources effectively, and don't pull more data than you need. If you have a lot of data, consider leveraging Performance Analytics for a faster, more scalable solution.

---

4. Common Reports You Might Need to Build from Scratch

Here are a few examples of reports you might need to rebuild, or reimagine, when moving from Cherwell to ServiceNow:

A. Service Desk Performance Reports

- In Cherwell: Reports that show how many incidents were resolved over a period, broken down by severity or by technician.
- In ServiceNow: You'll probably want to use Performance Analytics to build a real-time dashboard that shows SLA compliance, average resolution time, and number of tickets closed. You can also build a trended report to track performance over time.

B. SLA Compliance Reports

- In Cherwell: A report that tells you how well you're meeting SLAs, broken down by priority or department.
- In ServiceNow: You'll want to use ServiceNow Reporting to pull SLA data from the SLA and Incident tables. ServiceNow can also track SLA breaches, so consider creating a visual report that shows SLA health over time and by service.

C. Incident Resolution Time by Severity

- In Cherwell: A report that tracks incident resolution time based on severity.
- In ServiceNow: You'll probably need to rebuild this with a Data Source that joins Incident and SLA tables. From there, you can create a Performance Analytics dashboard to track resolution times in real-time.

D. Change Management Reports

- In Cherwell: A report that tracks approved changes, the number of successful changes, and change failure rates.

- In ServiceNow: You'd use Change Management data in ServiceNow, building reports to track the change lifecycle, success rates, and change impact. You could also use Performance Analytics for trend tracking over time.

---

5. How Long Will This Take?

Alright, here's the real talk: it depends. If you've got a small set of reports, it might take a couple of weeks. But if you've got a lot of complex reports (with lots of data transformations), expect a few months of work. Here's a rough breakdown:

- Small Migration (10-20 reports): 1-2 months
- Medium Migration (20-50 reports): 3-4 months
- Large Migration (50+ reports): 6 months or more

So, buckle up.

Migrating reports from Cherwell to ServiceNow is like upgrading your phone—you know it's going to be better, but it's going to take a little time to get used to the new features. Don't expect everything to be perfect on day one, but with careful planning, solid testing, and a bit of patience, you'll get there. And hey, at least you're not stuck with those ancient Crystal Reports anymore.

Compliance Considerations: SOX and SOC Audits

Migrating from Cherwell to ServiceNow can have significant implications for compliance, particularly with **Sarbanes-Oxley (SOX)** and **Service Organization Control (SOC)** audits. To ensure continued compliance and avoid deficiencies during the transition, consider the following measures:

Key Compliance Risks During Migration

- **Data Integrity & Security:** Migrating sensitive financial or customer data without proper controls can introduce security risks.
- **Audit Trail Disruptions:** SOX and SOC require consistent and auditable records; data inconsistencies could result in compliance gaps.
- **Access Control Changes:** Moving to ServiceNow may require role and permission adjustments, potentially leading to unauthorized access risks.
- **Process & Workflow Alignment:** Automated approval workflows and change controls must be validated to ensure they meet compliance standards.

Best Practices to Maintain Compliance

1. **Perform a Compliance Risk Assessment**
    - Identify critical compliance-related data and workflows in Cherwell.
    - Map SOX and SOC requirements to corresponding ServiceNow functionalities.
2. **Ensure Proper Access Controls in ServiceNow**
    - Define **Role-Based Access Control (RBAC)** policies early in the migration.
    - Use **ServiceNow Access Control Rules (ACLs)** to enforce data security.
3. **Validate and Preserve Audit Trails**
    - Ensure historical audit records from Cherwell are migrated accurately.
    - Leverage **ServiceNow Audit Logging** and **Change History** for compliance tracking.
4. **Automate Compliance Monitoring**
    - Utilize **ServiceNow Governance, Risk, and Compliance (GRC)** module to enforce policies.
    - Configure alerts for **policy deviations, unauthorized access, or workflow failures.**
5. **Engage Compliance and Audit Teams Early**
    - Work closely with internal auditors and compliance officers to validate controls.
    - Conduct **pre-audit assessments** before finalizing migration.

By taking these measures, organizations can ensure that the migration does not introduce deficiencies, maintains compliance with SOX and SOC standards, and strengthens governance in the new ServiceNow environment.

Organization Change Management Considerations

1. Stakeholder Engagement and Communication Plan

- Goal: Get everyone on the same page and avoid "what the heck is happening?" moments.
- Tools: Stakeholder Matrix & Communication Plan Template
- What to Do:
  - Make a list of who needs to know about the migration (think: IT folks, leadership, and anyone who'll be affected by this switch).
  - Craft a communication plan that's not full of jargon. You want people to actually *read* the updates.
  - Send regular updates, even if it's just to remind people this is happening and they should probably care.
  - When? Start in the planning phase and keep updating folks until the whole thing's done. No radio silence!

2. Change Impact Assessment (CIA)

- Goal: Figure out who's going to hate this and who's secretly excited.
- Tools: Change Impact Assessment Template
- What to Do:
  - Talk to people from different departments (yes, even the ones who usually avoid meetings) to see how they'll be impacted.
  - Map out what changes—who's going to be confused by new tools and who's getting a nice upgrade.
  - When? Do this early on (first month), before anyone starts getting too attached to Cherwell.

3. Training and Knowledge Transfer Plan

- Goal: Make sure no one's asking, "How do I…?" at the last minute.
- Tools: Training Needs Assessment Template & Knowledge Transfer Plan Template
- What to Do:
  - Develop a training plan that covers all the basics and the fancy new features.
  - Don't forget: videos, manuals, and maybe some role-specific training for the super-nerds out there who want to dive deep.
  - When? Start planning early (around month 2), but don't roll out training until 2–3 months before go-live—gives people enough time to absorb it.

4. Change Champion Network

- Goal: Find those brave souls who'll help others get through the chaos without giving up.
- Tools: Change Champion Selection Criteria & Responsibilities Template
- What to Do:
    - Pick people who are respected and can explain things without sounding condescending.
    - Train them up so they can help others who might still be clinging to Cherwell like a security blanket.
    - When? Get these folks on board early (month 1–2), and start tapping them for help during training and post-go-live.

5. Resistance Management Plan

- Goal: Help people who would rather throw a tantrum than switch tools.
- Tools: Resistance Tracking Template
- What to Do:
    - Predict where the biggest complaints will come from (spoiler: it's usually *change* itself).
    - Prepare strategies for dealing with resistance—whether it's extra training, hand-holding, or just listening to complaints about "how things used to work."
    - When? Start tracking resistance early and keep tabs on it through the whole process.

6. User Acceptance Testing (UAT) and Feedback Loop

- Goal: Make sure ServiceNow doesn't just look good in theory, but actually works for real users.
- Tools: UAT Plan Template & Feedback Collection Template
- What to Do:
    - Get a diverse group of users (and yes, some of them will inevitably try to break it) to test the system.
    - Collect feedback and *actually* listen to it (you'll be glad you did later).
    - When? Start UAT a couple of months before go-live, and keep the feedback loop going post-launch—people love to complain about the "new system."

7. Post-Go-Live Support and Adoption Monitoring

- Goal: Keep the wheels turning once the shiny new system is live.
- Tools: Post-Go-Live Support Plan Template & Adoption Metrics Dashboard
- What to Do:

- Have a team ready to handle questions (because there will be questions) and address any hiccups.
- Track how well people are adapting—if they're still confused by the basics, you may need to add more training.
- When? The first 3–6 months post-go-live are crucial. You want people to stay engaged and not just start using Excel again.

8. Change Management Metrics and KPIs

- Goal: Measure whether all your hard work is paying off or if you've just been spinning your wheels.
- Tools: Change Management Dashboard Template & KPIs for Change Adoption
- What to Do:
  - Track KPIs like how many people are actually using ServiceNow, training completion rates, and any resistance points.
  - Adjust your plan if you notice any red flags—no one likes a failed adoption.
  - When? Track these from day one, and especially after go-live, so you can fix things quickly.

9. Go Live and Recognition

- Goal: Celebrate the fact that you survived, and maybe even *thrived*.
- Tools: GoLive Plan Template
- What to Do:
  - Recognize people who helped make the migration a success (because they deserve it).
  - Throw a GoLive celebration (virtual or in person) so everyone can take a deep breath and realize this nightmare is over.
  - When? Celebrate key milestones—like the fact that no one screamed when the system went live.

Timeline Recap:

1. Initial Planning (Month 1-2): Stakeholder communication, impact assessment, and finding your change champions.
2. Training & Development (Month 3-5): Designing and rolling out training, resistance management, and making sure your champions are ready.

3. Testing & Pre-Go-Live (Month 6-7): UAT, feedback collection, and preparing for the inevitable questions.

4. Go-Live & Post-Go-Live (Month 8-12): Support, monitoring adoption, and making sure people don't go back to their old ways.

Baseline Project Plan

Phase 1: Planning & Assessment (4-6 Weeks)

- Define project scope, objectives, and success criteria.
- Conduct a detailed assessment of Cherwell environment.
- Identify key stakeholders and assign roles.
- Develop risk mitigation strategies.

Phase 2: Design & Preparation (6-8 Weeks)

- Finalize ServiceNow module selection.
- Define data migration and integration strategy.
- Establish governance and compliance checkpoints.
- Begin initial user training sessions.

Phase 3: Migration & Implementation (8-12 Weeks)

- Extract, transform, and load data into ServiceNow.
- Recreate necessary workflows and automations.
- Set up integrations with third-party systems.
- Conduct iterative testing and validation.

Phase 4: User Training & Go-Live (4-6 Weeks)

- Deliver final user training and hands-on workshops.
- Conduct a soft launch with a pilot group.
- Address user feedback and fine-tune configurations.
- Official go-live and post-launch monitoring.

Phase 5: Optimization & Continuous Improvement (Ongoing)

- Monitor system performance and user adoption.
- Optimize workflows and automation rules.
- Implement feedback-driven improvements.

- Conduct periodic compliance and security audits.

Top 10 Reasons Why Cherwell to ServiceNow Migrations Fail

Here's the top 10 reasons why Cherwell to ServiceNow migrations tend to fail:

1. No One Knows Why We're Migrating

- Why it fails: You'd think it's obvious, right? You're migrating to something better. But sometimes, people dive in without really knowing why they're doing it. Is it just for "new and shiny" or is there a solid reason, like better reporting or scalability? Without a clear mission, you're just spinning your wheels.
- How to fix it: Define your goals. Are you looking for performance improvements? Integration? Better reporting? Get everyone on the same page—otherwise, this migration will end up like every project with no clear vision: a hot mess.

2. Underestimating How Hard This Actually Is

- Why it fails: People think migrating from Cherwell to ServiceNow is just a "plug-and-play" operation. It's not. It's like assuming you can just move into a new house without a million boxes of stuff to organize. You've got to manage data structures, workflows, customizations, and a bunch of other things you didn't even know you had.
- How to fix it: Get real about the complexity. ServiceNow is powerful, but it's not a free pass. Don't go it alone—get experts who know how to navigate both platforms. Mana'o Pili has the experience and deep knowledge to help with data migration, customizations, and integrations. Whether it's setting up workflows or ensuring your post-migration support is handled, we've got you covered so you don't have to figure it out on your own.

3. Ignoring Data Mapping (Who Needs That?)

- Why it fails: It's one thing to migrate data; it's another to understand how that data is structured in both systems. If you don't map it out correctly, you're going to end up with a mess of data that doesn't make sense. And no one wants that. Trust me.
- How to fix it: Map your data. Yeah, sounds boring, but it's a must. Understand how data lives in Cherwell and how it should be structured in ServiceNow. Don't assume it will all just magically work. (It won't.)

4. Overlooking Customization Nightmares

- Why it fails: Both Cherwell and ServiceNow are like giant customizable Lego sets—endless possibilities, but also room for things to go horribly wrong. If you've got a ton of custom workflows and fields in Cherwell, you're going to have to redo them in ServiceNow. And that's not as fun as it sounds.
- How to fix it: Sit down and document your customizations. Make sure you know exactly what you're working with. Then figure out how to rebuild them in ServiceNow without losing your sanity.

5. Testing? Nah, Who Needs It?

- Why it fails: Who needs testing, right? You just "migrate" and hope everything works? Spoiler alert: it won't. Skipping testing is like driving a car without checking the brakes. Sure, it works—until it doesn't.
- How to fix it: Test everything. And don't just run a few sample reports and call it a day. Test every. single. thing. Get real feedback, especially from the people who will use this every day. If they're not happy, you've got problems.

6. No One Talks to Anyone

- Why it fails: If your stakeholders are out of the loop, you're done for. People who are actually going to use the new system (aka, the people who can tell you if this migration is even worth it) need to be involved from the start. Otherwise, you'll hear "Why did you move to ServiceNow again?" on day one of the rollout. And no one wants that.
- How to fix it: Talk to people. A lot. Especially your end users. They'll let you know if you're missing the mark, and trust me—they'll let you know if you didn't.

7. You're Understaffed, Undertrained, and Overwhelmed

- Why it fails: Migrating takes resources, and not just the "here's your budget and go" kind. You need expertise, and you need people—ideally ones who know both platforms. If your team is already doing 100 other things, don't expect them to have time for this.
- How to fix it: Don't be stingy with resources. Get some trained people on board. Hire external experts if needed. This isn't a side project; this is a big deal.

---

8. Data's Not As Clean As You Think

- Why it fails: If your data is dirty, it'll mess up everything in the migration. Think about it—would you want to move all your junk to a new house without sorting through it first? No one wants a duplicate mess in ServiceNow. It's like inviting clutter to the party.
- How to fix it: Clean your data before migration. Do a full audit, and get rid of anything that doesn't need to be there. It'll save you so many headaches later. Trust me.

9. People Hate Change (And They Will Complain)

- Why it fails: Humans. They don't like change. Especially when it's forced upon them without explanation. If you don't manage the transition, your users will be miserable. And miserable users are loud users.
- How to fix it: Implement solid change management. Talk about the benefits of ServiceNow. Train your users. Set expectations. Give them a reason to love the new system (or at least tolerate it).

10. Post-Migration Support? Nah, We're Good

- Why it fails: Once the migration's done, people think it's over. Guess what? It's not. If you don't have a support plan in place for post-migration, you're setting yourself up for a disaster. Stuff breaks, users need help, and things go sideways—and without support, you're out of luck.
- How to fix it: Have a support plan. Get a team ready to fix things as they come up. It's not going to be perfect the first time around, so be prepared for things to break (because they will).

Migration Failure Summary

Migrating from Cherwell to ServiceNow isn't some easy task that's going to magically happen because you've got a few PowerPoints. It's a massive process that requires planning, testing, resources, and communication. So, don't go into it thinking it's all rainbows and butterflies. It'll take work, but if you dodge these common mistakes, you'll have a much smoother ride. Happy migrating!

Conclusion

Migrating from Cherwell to ServiceNow is an opportunity to improve IT service delivery, enhance user experience, and future-proof your organization. By following best practices, ensuring compliance, and leveraging the right ServiceNow modules, organizations can achieve a smooth and secure transition.

Bibliography

- ServiceNow. (2024). *IT Service Management Best Practices.* Retrieved from www.servicenow.com
- Cherwell Software. (2023). *Cherwell Service Management Documentation.* Retrieved from www.cherwell.com
- ITIL Foundation. (2022). *Best Practices for IT Service Management.* Retrieved from www.axelos.com

About Mana'o Pili ↘

Mana'o Pili is a Hawai'i based technology consulting firm specializing in business automation through ServiceNow. Mana'o Pili provides its customers with individualized solutions. We reject the notion of one-size-fits-all solutions. Instead, we partner with you to craft a tailored plan that aligns with your unique needs, budget, and objectives. Our approach focuses on optimizing your existing platform while minimizing customization and reducing technical debt.

Here at Mana'o Pili, we treat our customers as 'ohana (family), listen closely to your challenges and deliver tailored attention with exceptional service.