PLAT4MATION

Whitepaper

Everyone loves vanilla

ServiceNow instance strategy:
From Heavily Customized to Plain Vanilla

Contents

Introduction

ServiceNow is known for its high degree of configurability and compatibility with hundreds of other applications. Nowadays, it is ideally suited voor IT Service Management (ITSM), and is well suited to solve bottlenecks for IT operations management, cloud management, security management and for example governance, risk and compliance. When ServiceNow was first released, a lot of the current functionality was not developed yet. As a result, customers of ServiceNow (usually their implementation partner) added a lot of custom developed functionality to meet their demands. Some implementation partners pre-packaged the custom developed functionality in what we call 'vendor packs' to speed up implementation timelines.

These so called 'vendor packs' are still in operation at many enterprises. On top of this, additional custom functionality was added to ensure a perfect fit to client processes. This has resulted in ServiceNow instances that have become large webs of custom code, additional custom applications and custom API's that need to work together inside the platform, but require large amounts of additional coding to be able to do so. Not surprisingly, technical debt has become a self propelling and elusive problem, because each fix creates new issues that need additional coding.

Because of all the additional coding and resulting technical debt, these old instances of ServiceNow are often perceived as slow, buggy and difficult to change, so they don't fully meet the requirements of their users anymore. Also the ability to keep up-to-date to new releases of ServiceNow is negatively impacted, so customers are confronted with high upgrade costs and long lead times. Last but not least, there are additional (concealed) costs because productivity of employees is impeded and developers spend a lot of time fixing avoidable problems.

The dilemma enterprises with these types of ServiceNow instances face is that they can't retire it outright without jeopardizing IT operations. Ideally, they would start all over again with a 'vanilla' (or new, unmodified) instance of ServiceNow, especially since current releases incorporate almost all of the functionality any enterprise may desire.

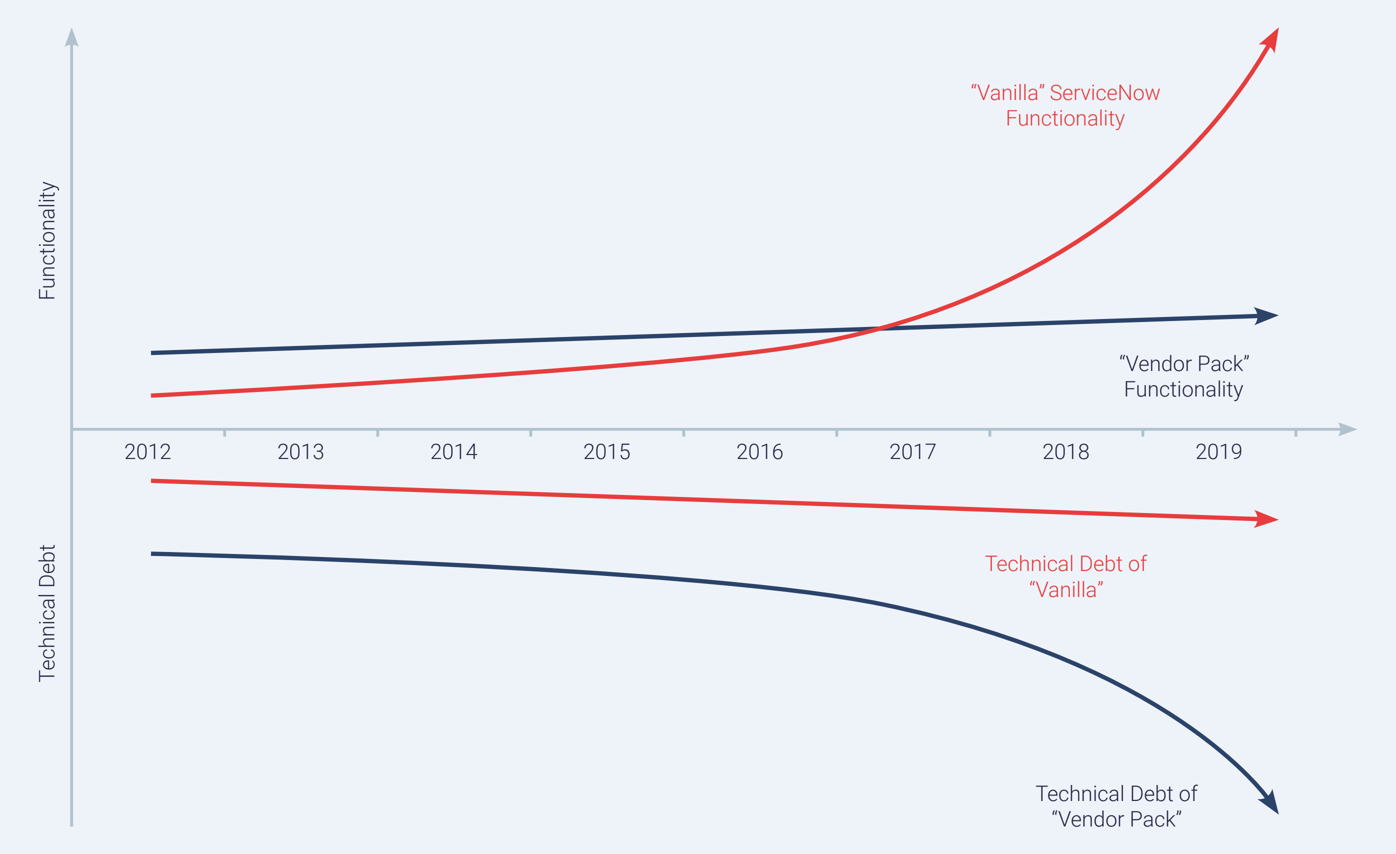
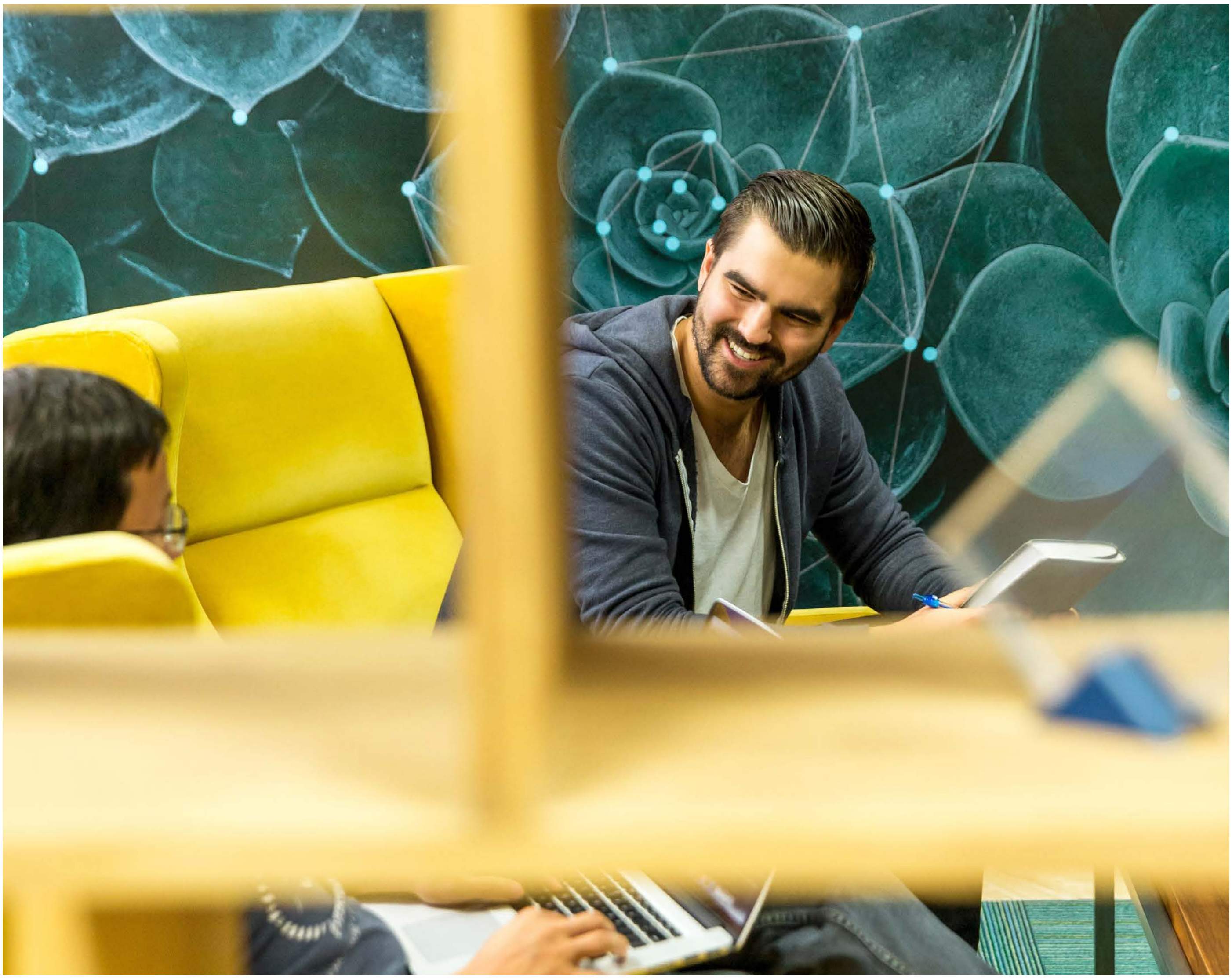


Figure 1 - Technical debt vs. functionality



Replacing old instances of ServiceNow is however quite challenging. IT operations and service management cannot be disturbed. Users need to be persuaded that migration is in their best interest. And all data in the old instance needs to be transferred smoothly to the new instance, without inheriting any technical risk or debt.Next to that, the new instance should offer all the functionality of the old one in a more user friendly way.
This whitepaper will present a step by step strategy to successfully migrate to a vanilla instance of ServiceNow. Especially in large organizations that need to comply to (external policies) and rely on a complex (outsourced) application landscape, it is key that the new instance can be fully (ISO) controlled and is properly integrated.

We will first take a look at three different scenarios to move away from heavily customized 'vendor pack' instances. After our choice for a scenario has been made, we will discuss the best course of action for migrating users as well as IT services, from infrastructure all the way up to the servicedesk. After our strategy and plan of action have been determined, we will take a look at the migration itself. We will introduce four principles that are critical to a successful migration. Finally, we will discuss how all essential data, integrations and functionality is successfully migrated without technical risk or debt using our integration Application called Connector4U.

Vendor packs of ServiceNow are often riddled with technical debt.

Chapter 1.
Three transition scenarios to move away from heavily customized instances

Is a vanilla instance really necessary? Isn't it possible to rebuild the existing instance, in effect creating a vanilla instance? Of course, it would be a valid choice to not migrate to a new vanilla instance at all. But the benefits of migration far outweigh the drawbacks. This chapter provides you the rationale behind this conclusion.

> **Vendor packs of ServiceNow are often riddled with technical debt.**

Especially for larger organizations and on the enterprise level, the newest release of ServiceNow offers far more than ITSM alone. The platform can be used as the cornerstone of IT and business operations, ranging from the already mentioned ITSM to an enabler for DevOps, Agile at scale, facility management processes and onboarding of new employees or customers. ServiceNow nowadays can help to make almost all employees more effective, regardless of the job they are doing.

Any organization could benefit from having a vanilla ServiceNow instance. The question is: how to make sure no data is lost and users will experience no hindrance during the transition period to a vanilla instance? While the transition to a vanilla instance is complex, it has the huge benefit of providing users with the desired functionality, a better user experience and better utilization of resources because developers can focus on other, more beneficial projects.

We distinguish three transition scenarios:

1. **Rebuild vendor pack**
2. **'Big Bang' transition**
3. **Gradual transition**

	1. Transform Existing Vendor Pack Instance	2. Gradually Phase-in Vanilla	3. Big Bang to Vanilla
Functionality Start vs Target	◑ Expected that re-factoring will not lead to full match with target	◕ Initially, users will need to use 2 systems, reducing the functional fit	● Expected that a pure OOTB solution will be more in line with target
Organizational Effort	◕ Users understand current setup and Vendor Pack already embedded in Customer infra	◑ User need to migrate to new system, but proposed to do phased approach	◔ Big bang approach will have large impact on people and systems
Effort to Integrate	◕ No additional integrations required compared to other scenarios	◔ Eventually Vendor Pack integration has to be rebuild, temporary integration with vanilla is needed	◑ Required to re-build (some) Vendor Pack integrations
Effort to Change	○ Re-factoring will lead to more code, thus making the solution more complex	◕ Vanilla / OOTB functionality will be easier to change	◕ Vanilla / OOTB functionality will be easier to change
Effort to Build	○ The CMDB is at the root of ServiceNow. A lot of work is expected to re-factor solution	◑ Building CMDB based on OOTB has low impact, however eventually ITSM also needs rebuild	◔ Rebuild of all functionality will have large impact
Effort to Maintain	○ Also re-factoring will add to Vendor Pack code base, adding to maintenance complexity	◕ Temporarily 2 systems need to be maintained adding to maintenance costs	● Big bang will lead to 1 system that needs to be maintained. Vanilla has lower maintenance
Benefits realization	◔ Deviation from OOTB will lead to lower benefits, re-factoring will lead to later benefits	● Quick benefits based on OOTB CMDB unctionality	◑ Late expected benefits
Overall	◔	◕	◕

● High fit ○ Low fit

Figure 2 - Three scenarios for migration to vanilla instance including benefits & drawbacks

Rebuilding with broken parts results in something that is still broken

The first option listed, is the scenario where the existing 'vendor pack instance' is re-build to align to the ServiceNow Out-Of-The-Box (vanilla) functionality. We consider this scenario the least desirable. Vendor pack instances of ServiceNow typically have huge amounts of technical debt. They are oftentimes beyond rebuilding because every change results in more technical debt, thereby achieving exactly the opposite of what you want. Even if one would try and somehow achieves the improbable by rebuilding the instance to 'vanilla', you will have spent an enormous amount of time and resources. Rebuilding the vendor pack is highly likely to not deliver the desired results and/or is very costly.

Nonetheless, it is still a scenario that some organizations consider or adopt. Rebuilding the vendor pack may be seen as the least risky scenario. Although we do not recommend this scenario, management of an organization may resist what is perceived as a dangerous overhaul of a critical system (we will touch on this issue in chapter 2) and thus have a reason for rebuilding old vendor pack instances. One has to keep in mind, however, that the overall effort will be very large, while the benefits are small compared to the other two scenarios listed. Last but not least, it is very probable that rebuilding the old vendor pack will not deliver the desired results and may even cause new problems.

Going out with a bang?

One of the other two options is what we like to call the 'Big Bang' transition. As the term suggests, the old instance would be retired outright while simultaneously kickstarting a new one. There are some important benefits to this approach: the complete user base will be migrated at once, all required functionality is immediately available, changes are easy to implement because there is no technical debt, and with a little effort it is possible to keep it that way. Finally, the maintenance effort is greatly reduced because technical debt has been eliminated.

There are some drawbacks however that may have organizations consider a gradual migration instead. First and foremost, a 'big bang' migration can be very disruptive for an organization as people and systems are impacted by the sudden change in functionality of the ServiceNow instance they use. The increased go-live risks associated to processes not working, compliance issues, integrations not working, etc., usually come with a higher demand from users to include all kinds of features that are currently in use in the old instance. It is highly probable that the increased risk is mitigated by adding more functionality, which will immediately lead to increased technical debt and moving away from vanilla functionality. Go-live dates could be postponed several times and as a result confidence levels in the new application will decrease. Overall, chances for organizational resistance are quite high in this scenario, which may jeopardize a successful migration and obtaining the initially sought benefits.

Easy does it

So, 'big bang' migration has some important advantages but also some major drawbacks. Isn't it possible to achieve a better balance between advantages and drawbacks by choosing a different approach?

The migration of a ServiceNow vendor pack instance to a vanilla instance doesn't have to be instantaneous. In fact, a gradual migration not only allows users more time to adapt to the new instance (causing less resistance), it also reduces the risk of organizational disturbances and functionality will be available sooner in comparison to a big bang. The old instance and the vanilla instance will for a while operate simultaneously, allowing for a smooth migration of critical data and user groups.

Conclusion

Our investigations into the pro's and con's of all three scenarios reveal that the "easy does it" scenario outperforms 'Big Bang' and 'Rebuild'. So our advice is to migrate gradually, but the question then becomes how this can successfully be done? Which steps need to be taken, how do you make the two ServiceNow instances 'talk' to each other, how can users be involved in a constructive way and at what point has the migration been successful? The next chapters will elaborate on these and other questions.

Chapter 2.
Haste makes waste: migrating the right users at the right time

Cementing support from users and migrating them at the right time are essential.

When opting for a gradual migration, two factors are crucial: ensuring the migration of data, integrations, functionality and users in the right order without incurring technical debt and cementing support within the organization. We will first take a look at the technical side of a gradual migration before discussing why support of user groups is vital to a successful migration.

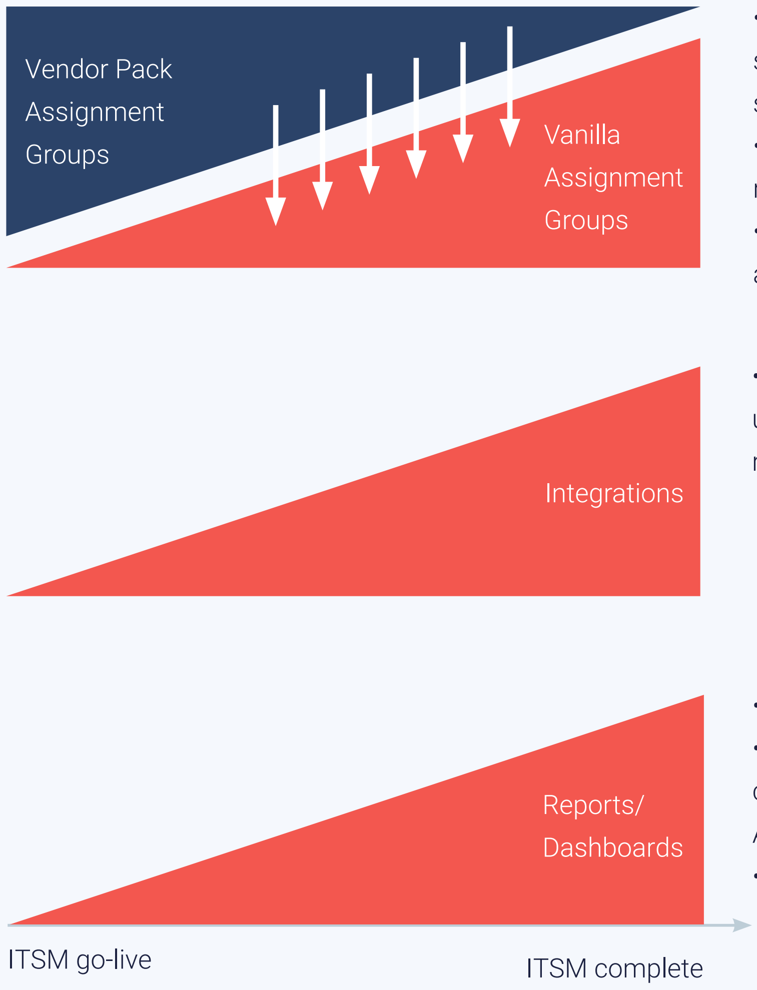
Operating two instances

As soon as basic ITSM functionality (CMDB, incidents, changes, requests) is available, users can be migrated to the new ServiceNow instance. It is strongly advised to do this in reverse order from a user perspective: the least user facing services are migrated first, the servicedesk is migrated last. If you would migrate the servicedesk first, the vanilla instance will immediately have to cope with high workloads, not in the least because the servicedesk uses a lot of additional functionality (e.g. call management). This can jeopardize a successful migration.

Usually, infrastructure services are migrated first, then application services, workplace services and finally the servicedesk. In this way, increasingly heavy ITSM workloads are gradually introduced to the vanilla instance while required functionality can be implemented in a structured way. The big advantage of moving smaller groups into vanilla is that the risks are lower compared to Big Bang, therefore resistance to change will be lower as well, which results in a lower need for having a perfect system with a lot of custom configurations. This will limit the amount of technical debt that will be accrued, compared to the Big Bang scenario.

It is essential during the transition that ALL ITSM data remains available in the 'vendor instance'. This data will be used in reports to prove compliance and will trigger the integrations that are currently in place (and may not be yet in place in the vanilla environment). During the migration, the data present in the vanilla instance is there to support the users that have been migrated. So for example, not the entire CMDB needs to be present in the vanilla instance during the transition period. During the transition, more and more compliance reports and interfaces will be implemented to ensure that once the servicedesk is migrated, a fully operational (close to vanilla) instance is in place.

After all processes, integrations and reports are live on the new vanilla instance, the old 'vendor pack' instance can be retired.



- Asignment Groups are migrated to Vanilla starting with 3rd and 2nd line, ending with the service desk
- Assignment Groups with heavy integration and reporting needs will be migrated later
- Agile/iterative approach provides time to learn and improve the transition

- Gradual build-up of all required integrations using Connector4U to ensure quick delivery of robust integrations

- Implement out-of-the-box reports
- Add additional reports/dashboards to support contractual/compliance needs and individual Assignment Group needs
- Growing number of reports/dashboards

Figure 3 - Gradual migration of assignment groups determined by the criticality of the business service they support

Champions of migration

Technical challenges aside, the second complex aspect of migration to a vanilla instance of ServiceNow are the people involved. It's no secret people are generally resistant to change. That's OK, after all it's just how most of us are wired. But it also is a significant variable in any migration from an old ServiceNow instance to a new one. The reason why people are so important for migrations is that many departments in an organization are dependent on ServiceNow. Although it probably doesn't work as it should or is desired, the people working in these departments have become accustomed to it.

Before anything is set in motion, it is therefore important to get the support of a group of key users that are enthusiastic about the proposed changes. Whenever possible, 'champions' need to be identified and engaged. For any given organization at least 10% of the users should commit to achieving the project objectives to ensure its success.

Chapter 3.

4 critical success factors

At the moment the first user groups are migrated, basic functionality has to be available. First, a configuration management database needs to be available to allow further development of the vanilla instance. Next, ITSM can be implemented so incidents and requests can be processed. After that, more user groups are migrated, all while the old instance is fully in operation, including the possibility to generate reports and overviews and integrations. A bypass is created to ensure both instances are working with the same data. To do this in the right way, four principles for success are essential:

- **100% possibility to generate all required reports from old instance**
- **All existing Integrations remain functional**
- **The service portal and servicedesk are migrated as last**
- **Communications with the customer is done from the old instance until Servicedesk is migrated**

Suffice to say, without the ability to generate reports the organization will be unable to gather essential information about daily operations to ensure it runs smoothly. The ability to generate reports needs to be maintained during the migration, not only to reduce organization resistance, but also to ensure the organization can prove they are compliant to internal (ISAE / ISO) and external policies (SOx).

Next, given the interconnectedness of functionality in the old ServiceNow instance to surrounding applications, whether it is via an integration that came with the vendor pack or as an integration that was added later on, it is essential all integrations remain functional. This is largely for the same reason a full capability for generating reports needs to be maintained: operations should be impacted as little as possible by the migration.

As we discussed in the previous chapter, the servicedesk and service portal are migrated last to enable the vanilla instance to be build gradually. As the migration progresses, the new instance of ServiceNow will duplicate more and more tickets, changes and requests that are processed in the old instance. Ultimately, at the moment the servicedesk and service portal are migrated, these user groups should not notice that they are working in a new instance of ServiceNow, except of course for a far better user experience.

Through all stages of the migration, it is essential to be transparent about the schedule, what the project means to different user groups, how they may be impacted, which measures are taken to ensure the impact is as small as possible and also ensure the benefits of the project are properly understood. After all, the whole endeavour is aimed at improving the organization, which is for an important part achieved by offering users a better experience working with ServiceNow.

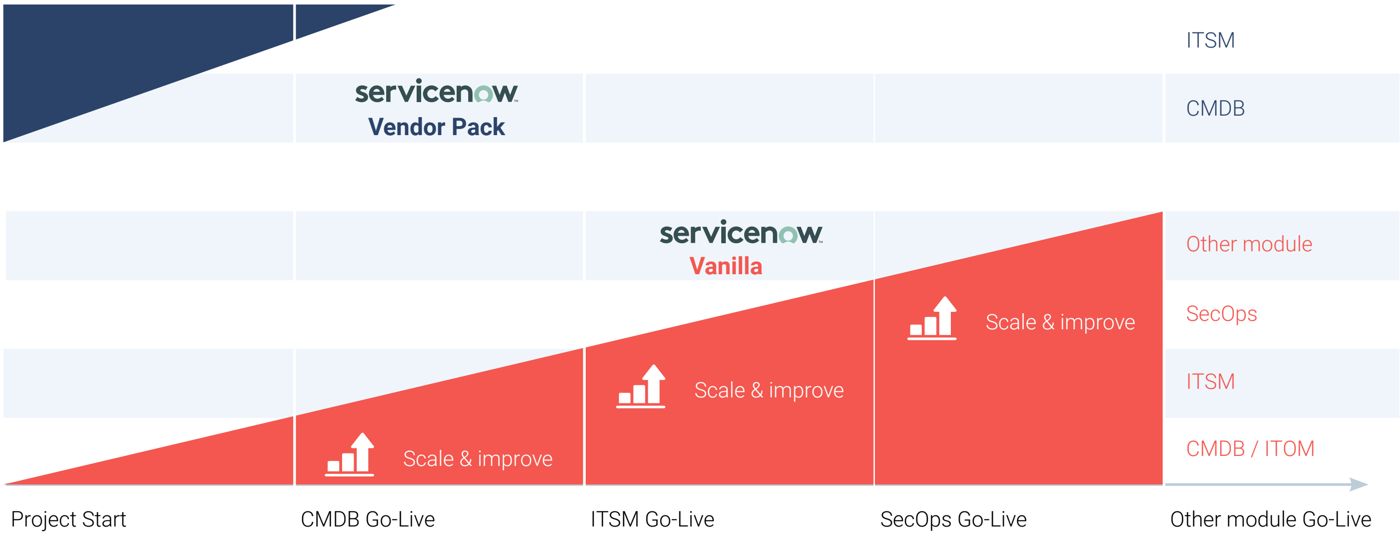


Figure 4 - Porting relevant functionality to the vanilla instance, starting with the CMDB

Chapter 4.

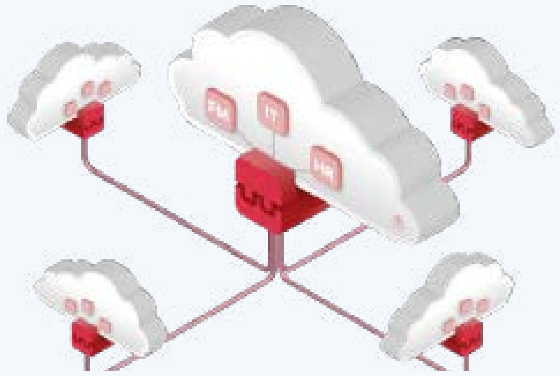
Reduce technical risk with Connector4U

One thing that has not yet been discussed is the question how a seamless integration between the old and new instance is achieved during the transition period. This is a challenge not to be taken lightly: it is absolutely crucial that the old vendor pack instance can exchange data with the vanilla instance, maintaining data integrity. Without the right tooling, this is impossible to achieve in a cost effective way. Efforts to do so without understanding how, can result in even more technical debt the organization set out to eliminate in the first place. To tackle this issue, Plat4mation has developed the integration application called Connector4U to ensure both instances maintain the desired functionality.

With Connector4U, the old instance of ServiceNow must be able to 'talk' to the new one to be able to migrate data. The vanilla instance at first acts like a ghost: it duplicates a growing amount of data (including tickets) from the old instance and is geared with dashboards and API's without feeding this back to the old instance. Users can initially work in the old instance; they will never have to work in two instances at the same time. As described in chapters two and three, a growing amount of user groups is gradually migrated to the new instance and works from there, while Connector4U ensures they retain access to the data in the old instance which is still being used by their colleagues.

Conclusion

Migrating to a vanilla ServiceNow instance can be quite challenging. With this whitepaper we hope to have shown that with the right approach and by choosing the most viable scenario, there are important benefits to migrating to a vanilla instance. When an gradual approach is chosen, user groups are properly involved and convinced of the benefits of the project, and crucial principles of success are adhered to, the only thing that stands in the way of a successful ServiceNow migration to a vanilla instance is technical risk and a perfectly executed transition strategy. Fortunately, with Connector4U, this risk is reduced to a minimum while your transition to a vanilla ServiceNow instance is realized.



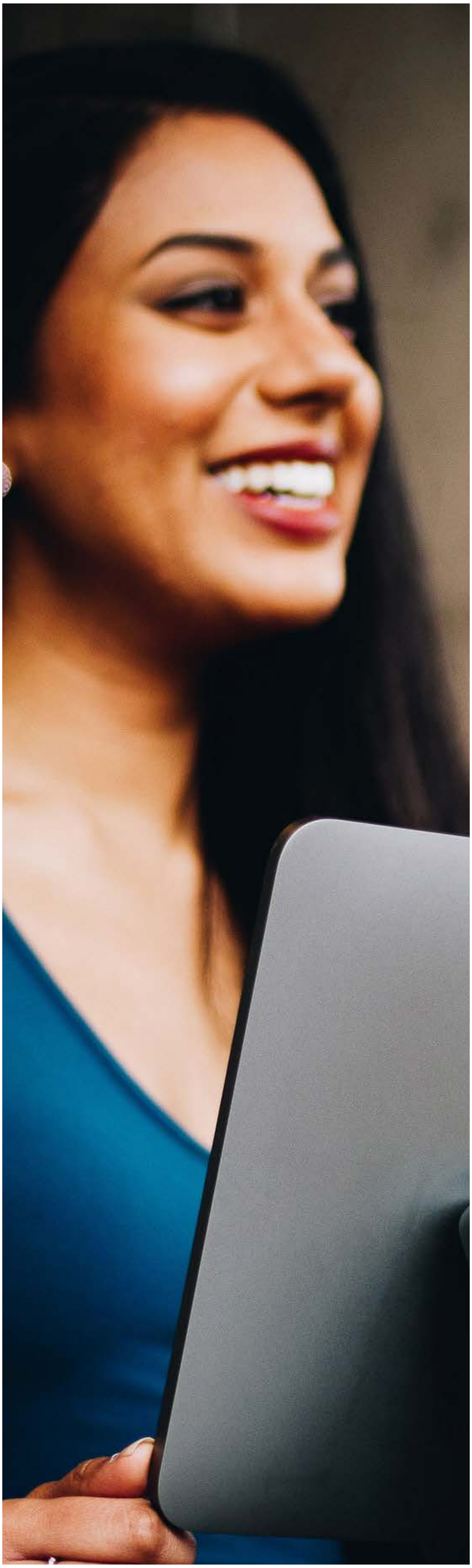
- **Connector4U can be easily installed within ServiceNow, no need for additional infrastructure or software licenses**

- **Easily create robust integrations that can handle complex business logic, using drag 'n drop.**

- **Connect to other ITSM Applications (ServiceNow and others) to exchange information related to incidents, changes, CI's, etc.**

- **Quickly build integrations to core (ERP) Applications, such as HR (Workday, Peoplesoft, SuccesFactors), Finance, Facility, etc.**

Figure 5 - Connector4U is used to quickly build integrations with ITSM applications such as ServiceNow



About Plat4mation

Based in The Netherlands, Belgium, Germany, USA and India, we are an Elite ServiceNow Partner dedicated to delivering world-class products and services for the ServiceNow platform. We are driven to realize maximum value in the IT, employee, and customer workflow experiences for each one of our customers. We do this by providing a flawless customer experience utilizing our extensive expertise.

Since our inception in 2013, we have grown to more than 200 employees globally, and we are still growing strong! With a team of specialized consultants, we aim for the highest possible results while creating jaw-dropping experiences for our customers.

Authors

Elmer de Valk - Managing Director
elmer.de.valk@plat4mation.com

Contact

Plat4mation BV
Arthur van Schendelstraat 650
3511 MJ Utrecht, The Netherlands

+31 30 760 26 70
info@plat4mation.com
www.plat4mation.com