

TECHNICAL SCIENCES

COMPARATIVE ANALYSIS OF ERP SYSTEMS: MICROSOFT DYNAMICS NAV AND ODOO COMMUNITY

Sretenović A.,

Graduate Engineer of Organizational Sciences

Pantelić O.,

Associate Professor, Faculty of Organizational Sciences

Đukić M.

Teaching Associate, Faculty of Organizational Sciences

<https://doi.org/10.5281/zenodo.10082538>

Abstract

The paper provides a detailed comparison between two widely used ERP systems, Microsoft Dynamics NAV 2017 and Odoo Community, focusing on their performance within a Project Management scenario. The goal of the research is to evaluate and contrast the functionalities, scalability, and customization capabilities of each system, particularly in relation to their suitability for project and service management. The paper emphasizes the significance of thoroughly assessing organizational requirements before selecting an appropriate ERP system, highlighting the crucial role of tailored decision-making in this process.

Keywords: *comparative analysis, Microsoft Dynamics NAV, Odoo Community, project management*

Introduction

In the context of a rapidly changing and unpredictable business environment, organizations encounter various obstacles in the pursuit of adaptation to emerging trends. To navigate such challenges effectively and attain a competitive edge, the adoption of ERP systems emerges as a top-tier solution (Leon, 2019). An ERP system represents a software tool that facilitates the integration, automation, and management of diverse business processes within a company. The fundamental objective of implementing an ERP system is to consolidate information and streamline operations across multiple departments, including finance, human resources, procurement, manufacturing, sales, and customer support. This integration aims to optimize decision-making, resource allocation, and overall operational efficiency within the organization [1].

Comparison of Microsoft Dynamics NAV 2017 and Odoo Community

A key distinction between Odoo Community and Microsoft Dynamics NAV 2017 software is their fundamental approach to software access. Microsoft Dynamics NAV 2017 operates under a licensed model, necessitating a fee for access and maintaining proprietary ownership. Users are unable to alter the system's source code, constraining its adaptability to meet specific requirements. On the other hand, Odoo Community is an open-source platform, meaning it is freely accessible and allows users to access and modify the system's source code. This open approach grants users significant flexibility and customization options, facilitating adaptation to individualized needs [3].

The system architecture of an ERP system defines the interconnection between its various components and modules, having a direct influence on its functionality, scalability, and support for organizational business processes. In the case of Microsoft Dynamics NAV 2017, the system architecture is structured around a three-tier model.

This model encompasses a client tier, the RoleTailoredClient, serving as the user interface; a logic tier, the Microsoft Dynamics NAV Server, handling application logic; and a data tier powered by SQL Server. Notably, this system primarily operates using the C/AL programming language [5].

Similarly, Odoo operates using a three-tier architecture. At the presentation tier, the user interface is built using HTML5, JavaScript, and CSS, ensuring a dynamic and interactive experience. The application logic is coded in Python, showcasing the system's flexibility and robust functionality. The data tier relies exclusively on PostgreSQL as the relational DBMS, ensuring efficient data management and storage. Notably, the system is organized into modules that facilitate the addition of new functionalities or the modification of existing ones. Modules visible to users are labeled as 'Applications,' while supplementary modules, referred to as 'addons,' are accessible via the `addons_path` [2].

Given that Microsoft Dynamics NAV 2017 is a licensed software, organizations are restricted from directly modifying the system's source code. Nevertheless, alternative ways for independent customization are available, including configuration, which enables the establishment of customized fields, tables, workflows, and permissions without necessitating alterations to the source code. However, substantial modifications to the system demand proficient programming skills. Microsoft Dynamics NAV 2017 employs C/AL as the programming language, which requires developers to possess specific expertise for efficient program development and maintenance. Notably, the integration of custom objects can complicate the migration process to new system versions and necessitate additional efforts during system updates [6].

Renowned for its adaptability and broad customization capabilities, Odoo is structured around a

modular framework that empowers organizations to tailor specific modules without causing system-wide disruptions. This modular approach proves particularly beneficial for businesses seeking to gradually introduce customized solutions or those with distinct requirements within particular business segments. Notably, Odoo leverages the Python programming language, recognized for its widespread popularity and accessibility, facilitating development accessibility for a diverse pool of programmers [2].

The concept of scalability within a system pertains to its capacity to accommodate increased demands in transaction volume, data storage, and the number of system users. In the case of Microsoft Dynamics NAV 2017, the system relies on Microsoft SQL Server as its database management system, renowned for its robust scalability features. SQL Server effectively manages large datasets and intricate queries, ensuring seamless operation even under increased workloads. Additionally, it supports the implementation of database partitioning, a technique that involves segmenting the database into smaller data groups, or partitions, which can function independently while belonging to the same logical database [8].

In contrast to Microsoft Dynamics NAV, Odoo Community operates using the PostgreSQL database, renowned for its specialized capabilities in managing large datasets efficiently. In addition to this, Odoo facilitates enhanced database scalability through its provision of fragmentation and partitioning options. While NAV's three-tier architecture enables scalability at each layer individually, Odoo, operating within a similar three-tier architecture, further extends its scalability to the module and application levels, thereby providing increased flexibility and adaptability within the system [4].

The interface of Microsoft Dynamics NAV is characterized by a conventional design featuring cards and data entry forms. Typically, these forms incorporate a menu situated above the data entry fields, providing access to numerous related options. While this interface is notably functional, the multitude of available choices can occasionally compromise clarity and intuitiveness. As a result, the learning process can be demanding, often requiring new users to undergo comprehensive training in order to effectively navigate and utilize the system.

In contrast, Odoo features a contemporary and intuitive interface characterized by a streamlined design. The main menu is strategically located on the left, facilitating convenient access to various applications and modules. Upon accessing a specific module, the interface presents only the essential options, leading to a more detailed exploration of functionalities. Key options are clearly highlighted, and the system incorporates helpful tools and pop-up notifications, contributing to an overall intuitive user experience. This modern interface design significantly enhances intuitiveness, compared to Microsoft Dynamics NAV.

Project Management module comparison

In the Project Management module of Microsoft Dynamics NAV 2017, the user interface serves as a

pivotal tool for efficiently creating, managing, and closely monitoring all projects and associated tasks. This interface offers a comprehensive view and control over every aspect of the project. Projects are presented in a conventional format, displaying relevant details in a list structure, including essential information like project name, current status, responsible person, and deadlines. However, this module currently lacks support for a Kanban approach. Consequently, visualizing the distribution of project phases, implementing color coding to signify project status, or defining key labels for enhanced search and organization poses a challenge within this system.

In the project creation process within Microsoft Dynamics NAV 2017, users input key details such as customer information, project duration, status, and billing method. Subsequently, they can define project phases and their associated tasks. A notable feature of this system is its ability to distinguish between billable and non-billable tasks. Tasks can be marked as 'Billable' to signify that payment is expected from the client upon the completed work. Conversely, tasks marked as 'Budget' indicate that they will not be invoiced, facilitating a clear differentiation between commercial and non-billable activities. Upon the successful completion of a project, the invoicing process is straightforward. Users can generate a Sales Invoice directly without the requirement for a prior Sales Order request. This approach significantly simplifies the invoicing procedure, contributing to the overall operational efficiency of the system.

Odoo provides an intuitive and well-designed user interface that equips users with a comprehensive set of tools and elements for effective management of diverse business aspects, including project management. The system's interface features a clean and organized layout, logical placement of elements, and user-friendly navigation, facilitating a quick grasp of its functionalities. Central to the project management interface is the Kanban board, where projects and associated tasks are presented as cards transitioning seamlessly between various stages. Each card contains key information, including project or task names, descriptions, deadlines, responsible team members, and current status. This visual presentation empowers users to swiftly assess project progress and identify key action items. A standout feature in Odoo's project management functionality is its ability to categorize projects. The system's flexibility allows users to utilize pre-defined project categories or create custom categories tailored to specific requirements. Furthermore, users can leverage color-coded projects and assign keywords as labels to signify essential characteristics, further enhancing usability and organization within the system.

A critical aspect of effective project management involves the definition of stages and their corresponding tasks. Implementing the Kanban approach facilitates a distinct definition between different project phases, thereby simplifying the tracking of task transitions and promoting streamlined team coordination and resource optimization. To document work data within a project, each task is

associated with Timesheets, allowing users to specify the assigned employee, the number of hours allocated to the task, and a detailed activity description. However, when billing a client for an employee's work, it is imperative to link each task with a predefined service provided by the employee; otherwise, the work will not be invoiced to the client. Upon inputting work data for all tasks in the project, users can access project-related statistics through various chart representations,

including histograms, line graphs, and pie charts. These charts can track parameters such as hours per stage, progress by stage, planned hours, overtime, and hours to project completion. Notably, immediate invoicing of a completed project to the client is not feasible; the creation of a Sales Order is a prerequisite for this process. A detailed comparison between the systems regarding project management is presented in Table 1.

Table 1.

NAV 2017 and Odoo Community differences in the Project Management module

Project Management module	Microsoft Dynamics NAV	Odoo Community
User interface and design	The intuitive user interface Logically organized elements Accessible navigation Traditional design	The intuitive user interface Logically organized elements Accessible navigation Minimalistic and modern design
Kanban board	Kanban is not supported All the essential project details are displayed in a list	Kanban is supported Projects and tasks are displayed as cards moving through stages
Project categorization	Not supported	Projects can be divided into different categories, coded with colors, and labeled with keywords
Stages and tasks	Available (without Kanban)	Available (through Kanban)
Billable and Non-Billable tasks distinction	Available	Billable tasks must be associated with a service listed in the Sales Order.
Invoicing	Direct creation of a Sales Invoice (prior Sales Order not requested)	Sales Invoice created upon the prior Sales Order

Service Management module comparison

Within Microsoft Dynamics NAV, the support for service contracts enables the establishment of contractual agreements with clients for product servicing purposes. Upon initiating a new contract, NAV offers several templates for straightforward contract creation. The contract defines essential details, including customer information, contract name, and the effective date. Furthermore, the contract amount is specified, alongside the selection of the billing and payment period. Notably, in the case of prepaid contracts, customers receive bills for each period, irrespective of whether services were utilized. Additionally, the contract necessitates the selection of the specific product to which the contract applies. Once the requisite data is inputted, the contract is saved and subsequently requires signing to finalize the agreement.

In the context of Service Management within NAV, the system facilitates the generation of Service Orders, triggered either by client requests or internal requirements. Users can provide a description of the servicing issue, utilizing predefined lists of errors, specifying the locations where they occurred, and outlining the symptoms associated with the occurrence. Each Service Order allows for the inclusion of multiple products necessitating servicing. For each line item within the order, users can specify whether it involves warranty service or is associated with a service contract. Subsequently, details regarding the required work and the responsible employee for the repair can be entered for each line item. The allocation of resources is facilitated based on their availability and defined working hours, with support for the tracking of resource usage, whether it pertains to human resources or parts. The system allows for the generation of invoices based on the services performed, with the

potential for automation to streamline the invoicing process further.

In contrast to NAV, Odoo Community does not provide direct support for service contracts. To emulate the functionalities of service contracts, users can leverage the Subscriptions application within Odoo. To begin, a subscription plan is created, specifying its name, the associated customer, and the designated billing or payment period. Subsequently, the establishment of a subscription is required. Notably, each subscription is linked to one or more products, necessitating the creation of the corresponding product within the system. Following this, the subscription plan is then associated with the intended service. The creation of a subscription involves the input of customer details and the specification of the subscription's validity period, followed by the selection of the service to which the subscription applies.

Service Management in Odoo software is available within the Repairs module. In this module, a repair order is created for a product or equipment. These orders include information about the product and the customer for whom the repair is performed, a description of the problem, and any relevant notes. However, each Repair Order refers to only one product for repair. After creating the Repair Order, it is possible to track the repair status, including monitoring different phases of the repair process, from the moment the product arrives for repair until its return to the customer. Invoicing can be generated based on Service Orders. To exclude charges for repairs, there is the 'No invoicing' option. The Repairs module can be configured to automatically generate invoices for the subscription period. Differences between the systems regarding servicing are presented in Table 2.

Table 2.

NAV 2017 and Odoo Community differences in the Service Management module

Service management	NAV 2017	Odoo Community
Service contracts	Supported Concluded with the client for servicing specific products	Not supported A similar effect can be achieved using Subscriptions
Service Orders	Service Orders created with the issue described through lists of errors and places of occurrence One Order can encompass multiple products for servicing	Repair Orders created with details about the product, customer, and issue Each Order refers to one product for repair
Resource planning	Definition of the required resources available Resources allocated based on availability	Repair progress tracking available Possibility of defining the required resources
Resource utilization	The resource usage tracking available	The resource usage tracking available
Invoicing	Invoice created based on servicing Possible automation	Invoice created based on servicing Possible automation

Conclusion

NAV 2017, as a commercially licensed solution, offers a wide range of features and integration with other Microsoft systems. It supports all the functionalities needed for efficient project management and has a service module that covers all forms of customer support. Odoo Community is an open-source software that provides a modern approach to business with flexibility and an intuitive user interface. Its modular architecture allows users to easily customize the system to their specific needs. Project management in Odoo is based on the Kanban approach, enabling visual project tracking. The service module provides basic functionalities but advanced options like service contracts are not available.

Microsoft Dynamics NAV 2017, operating as a licensed solution, provides an extensive array of features and seamless integration with various other Microsoft systems. It encompasses all the essential functionalities required for effective project management, alongside a comprehensive service module that caters to diverse forms of customer support. In contrast, Odoo Community serves as an open-source software solution, offering a contemporary business approach characterized by flexibility and an intuitive user interface. Its modular architecture allows users to readily customize the system to align with their specific requirements. Within Odoo, project management is facilitated through the utilization of the Kanban approach, enabling users to track projects visually. While the service module in Odoo provides fundamental functionalities, it lacks advanced options such as comprehensive service contracts.

Upon general comparison, it is evident that Odoo Community stands out as a more accessible and user-friendly solution, offering an intuitive interface and customizable options. However, when considering the breadth of functionalities and coverage of standard business processes, NAV 2017 emerges as a comprehensive solution. Within the scope of the presented modules, Odoo Community demonstrates its flexibility and adaptability in project management, while NAV 2017 showcases its strength in servicing capabilities.

The decision to opt for either the NAV 2017 or the Odoo Community system is contingent upon the unique needs and preferences of the organization in question. NAV 2017 is well-suited for medium and large-scale enterprises seeking seamless integration with other Microsoft systems, thereby streamlining overall operations. In contrast, Odoo Community caters to organizations that prioritize flexibility, particularly those employing agile project management methodologies that require adaptable and customizable solutions. By carefully assessing the specific requirements and priorities of the organization, businesses can make an informed choice that aligns with their operational and strategic objectives. Ultimately, the choice between the two would depend on the specific needs and priorities of the organization, considering factors such as user-friendliness, depth of functionalities, and customization requirements.

References:

1. Bradford, M. (2014). Modern ERP: select, implement, and use today's advanced business systems. Lulu.com.
2. Chapter 1: Architecture Overview — Odoo 16.0 documentation. (n.d.). Retrieved September 4, 2023, from https://www.odoo.com/documentation/16.0/developer/tutorials/getting_started/01_architecture.html
3. Ganesh, A., Shanil, K. N., Sunitha, C., & Midhundas, A. M. (2016, February). Openerp/odoo-an open source concept to erp solution. In 2016 IEEE 6th International Conference on Advanced Computing (IACC) (pp. 112-116). IEEE.
4. Jindal, N., & Dhindsa, K. S. (2013). Comparative Study of OpenERP and its Technologies. International Journal of Computer Applications, 73(20), 42-47.
5. Jswymer. (2018, January 2). Product Architecture - Dynamics NAV. Microsoft Learn. Retrieved September 4, 2023, from <https://learn.microsoft.com/en-us/dynamics-nav/product-and-architecture-overview>
6. Jswymer. (2022a, May 3). Personalizing pages in the Dynamics Windows Client - Dynamics NAV

app. Microsoft Learn. Retrieved September 4, 2023, from <https://learn.microsoft.com/en-us/dynamics-nav-app/ui-personalization-windows-client>

7. Jswymer. (2022b, June 9). Client Types - Dynamics NAV. Microsoft Learn. Retrieved September 4, 2023, from <https://learn.microsoft.com/en-us/dynamics-nav/client-types>

8. Jswymer. (2022c, June 9). SQL Server Database Components - Dynamics NAV. Microsoft Learn. Retrieved September 4, 2023, from <https://learn.microsoft.com/en-us/dynamics-nav/sql-server-database-components>

9. Leon, A. (2014). Enterprise resource planning. McGraw-Hill Education (India) Pte Limited.