# Entwickeln von Erweiterungen für Microsoft Dynamics NAV unter Verwendung von ExtensionsV2

(Exposé)

Johannes Naderer
(se0307)

# Kapitel 1

# Thema

In dieser Arbeit soll der Leser schrittweise an das Thema der Erweiterungsentwicklung im ERP-System Microsoft Dynamics NAV eingeführt werden. Das Thema der Erweiterungsentwicklung im Dynamics NAV Umfeld wurde erstmals mit Erscheinen der Version 2018 relevant. Für dieses neue Konzept wird seitens Microsoft die neue Sprache *AL* entwickelt, die zwar auf der bisherigen Sprache *C/AL* aufbaut, jedoch einige Einschränkungen und Erweiterungen mit sich bringt. Ziel der neuen Sprache und des Entwicklungsparadigmas rund um sie ist es, Entwicklern einen zeitgemäßen Entwicklungsprozess zu erleichtern und gleichzeitig bereits innerhalb der Sprache Vorkehrungen für den Sprung in die Cloud zu treffen.

# Kapitel 2

# Einleitung

## 2.1 Motivation

ERP-Systeme sind heute aus dem wirtschaftlichen Umfeld nicht mehr wegzudenken. [2] [1] Wo in den 1970er und 1980er Jahren innerhalb der einzelnen Abteilungen von Unternehmen verschiedene Software-Lösungen verwendet wurden, ist man sich mittlerweile einig, dass eine zentrale Applikation zur Verwaltung aller unternehmerisch wichtigen Daten und Prozessschritte große Vorteile liefert. Gleichzeitig hatte dieses Umdenken zur Folge, dass kleine Softwarehersteller ins wanken gerieten - da seitens der Wirtschaft allumfassende Software-Giganten gefordert wurden, die mit eingeschränkten Ressourcen und Branchenwissen nicht entwickelt werden konnten. Genau diese Großsysteme werden neben dem Platzhirsch SAP und einigen anderen Unternehmen auch von Microsoft mit seiner Dynamics Sparte geliefert.

Vor der Kaufentscheidung für ein ERP-System gilt es neben den finanziellen Aspekten auch herauszufinden, welche Systeme die bestehenden Prozesse des Unternehmens am Besten abbilden. [2] Und genau hier spielt die Anpassbarkeit und Erweiterbarkeit eines Systems eine zentrale Rolle. Systeme bilden oft Geschäftsprozesse auf ihre eigene Art ab, und zwingen so den Anwender bestehende Prozesse anzupassen, oder das System zu ändern. Gerade bei branchenspezifischen Abläufen muss hier meist die Logik des ERP-Systems geändert oder erweitert werden. Solche Änderungen sind jedoch meist teuer und zeitintensiv. Gerade der finanzielle Aufwand einer ERP-Einführung stellt für Unternehmen ein großes Risiko dar. Da der Großteil der großen Systeme darüber hinaus auch periodisch mit Updates versorgt werden, um gegenüber rechtlichen Änderungen flexibel zu bleiben, können sich Eingriffe in die Logik auch negativ auf die Updatefähigkeit des Systems auswirken.

Um die Aspekte der Erweiterbarkeit und Anpassungsmöglichkeit zu bedienen, entschied sich Microsoft in seinem ERP-System Dynamics NAV bereits in sehr frühen Versionen, zertifizierten Entwicklern freien Zugang zum Applikationscode zu gewähren. [3] So können diese die gesamte Geschäftlogik je nach Anforderung abändern, aushebeln und erweitern. Das hat zum Einen zur Folge, dass Entwickler mächtige Applikationen erstellen können, und sich diese direkt in das bestehende System integrieren lassen, allerdings kommen mit diesen Möglichkeiten auch Probleme auf. Je weiter die oft über Jahrzehnte verwendeten Systeme vom Standardlieferumfang von Microsoft abweichen,

desto aufwändiger, teurer und fehleranfälliger ist es diese Systeme mit monatlichen und jährlichen Updates zu versorgen.

Um dieses Problem anzugehen, und gleichzeitig zukünftige Entwicklungen kompatibel zur Cloud-Variante des ERP-Systems zu halten, wurde mit der Version 2017 erstmals das Konzept der Erweiterungen mit ExtensionsV1 eingeführt. ExtensionsV1 war ein neuer Ansatz für das ERP-System zu programmieren und hatte viele Vorteile, war jedoch in einigen technischen Details noch nicht ausgereift und konnte sich daher gegenüber der konventionellen prozeduralen Entwicklung im Dynamics NAV Umfeld nicht durchsetzen. Mit der in 2018 veröffentlichten Version – ExtensionsV2 – wurden nicht nur viele diese Mängel behoben, sondern auch eine neue Sprache mitsamt Entwicklungsumgebung vorgestellt.

## 2.2 Zielsetzung

Ziel dieser Arbeit ist es, dem Leser einen generellen technischen Überblick über die Architektur des Systems Microsoft Dynamics NAV zu geben. Dafür ist es notwendig, das Prinzip der verwendeten 3-Schichten-Architektur zu verstehen. Zusätzlich wird geklärt, wie und wo C/AL bzw. AL Code kompiliert und schlussendlich ausgeführt wird. Anschließend soll die erweiterungsbasierte Programmierung zum bisherigen Ansatz der traditionellen prozeduralen Programmierung verglichen werden. Dazu wird zuerst ein Einblick in die Programmierung in der C/SIDE Entwicklungsumgebung mit der Sprache C/AL an einem einfach gehaltenem Beispiel vermittelt. Danach wird dasselbe Beispiel mit der neuen Sprache AL in Visual Studio Code erstellt. In der anschließenden Diskussion wird dann im Detail auf die Gemeinsamkeiten und Differenzen beiden Arten eingegangen. Hierbei ist es nicht Ziel, syntaktische Kleinigkeiten hervorzuheben, sondern konzeptionelle Unterschiede, und deren Vor-/Nachteile zu beleuchten.

Der Leser soll nach Durchsicht der Arbeit einen Überblick über das Programmsystem Microsoft Dynamics NAV haben, und die beiden angewandten Programmierkonzepte verstehen und unterscheiden können.

# Kapitel 3

# Stand der Technik

Microsoft Dynamics NAV hat während seiner Entwicklung einige Architektursprünge hingelegt. Ausgehend von Navision für DOS zum Sprung als reine 2-Schichten-Architektur mit FAT-Client zur heutigen 3-Schichten Architektur. [4]

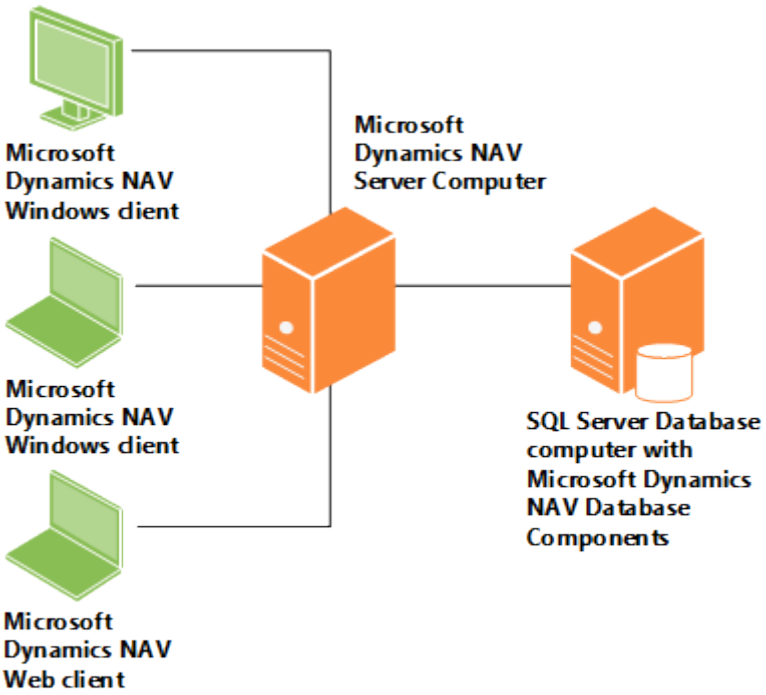


**Abbildung 3.1:** Overview: Dynamics NAV 3-Tier-Architecture https://docs.microsoft.com/en-us/dynamics-nav/product-and-architecture-overview

Abbildung 3.1 zeigt einen vereinfachten Grundaufbau der 3-Schichten-Architektur von Dynamics NAV. In Grün sehen wir die verschiedenen Client-Arten die die erste Schicht repräsentieren. Neben dem klassischen Windows-Client zählen auch der Web-Client und die verschiedenen Mobile-Apps zur Schicht Eins. Auf der Schicht Zwei befindet sich der NAV-Server. Dieser bildet den Kern ges Gesamtsystems, verbindet die Da-

tenbank mit den Clients und führt sämtliche Geschäftslogik aus. Die dritte Schicht bildet die Datenbank, auf der sämtliche Datenbestände des Systems verwaltet und persistiert werden. Als Datenbanksystem wird hier Microsoft SQL Server verwendet. Die Serverkomponenten (Server, Datenbank) können sowohl lokal installiert werden, als auch in die Cloud (Microsoft Azure) ausgelagert werden, wobei sämtliche Kombinationen möglich sind, und man daher freie Hand bei der Auswahl der Betriebsoptionen hat.

# Kapitel 4

# Geplantes Inhaltsverzeichnis

- Kurzfassung (1 Seite)
- Abstract (1 Seite)
- Einleitung (4 Seiten)
  - Motivation (2 Seiten)
  - Problemstellung/Zielsetzung (2 Seiten)
- Grundlagen (4-5 Seiten)
  - Grundarchitektur Dynamics NAV (2 Seiten)
  - Objektarten in Dynamics NAV (2-3 Seiten)
- C/AL Entwicklung (5-8 Seiten)
  - Vom Code zum Programm (C/AL Compiler Workflow) (1-2 Seiten)
  - Entwicklung im Development Environment (4-7 Seiten)
    - Syntax und Sprache (1-2 Seiten)
    - Beispielapplikation (3-5 Seiten)
- AL Entwicklung (7-11 Seiten)
  - Entwickeln in Visual Studio Code (2-3 Seiten)
  - AL - Sprache und Syntax (2-3 Seiten)
  - Beispielapplikation (3-5 Seiten)
- Diskussion und Analyse (4-5 Seiten)

# Kapitel 5

# Vorläufige Literaturliste

Literatur


1. Schwickert et. al: Einführung in MS Dynamics NAV 2009
2. Bernard Wong, Dr and Tein, David: Critical Success Factors for ERP Projects
3. A. Duplaga, Edward and Astani, Marzie: Implementing ERP in Manufacturing
4. Mark Brummel: Learning Dynamics NAV Patterns
5. Jörg A. Stryk: The NAV/SQL Performance Field Guide
6. Rabindra Sah: Mastering Microsoft Dynamics NAV 2016

# Quellenverzeichnis

## Literatur


[1] Edward A. Duplaga und Marzie Astani. „Implementing ERP in Manufacturing“. 20 (Juni 2003), S. 68–75. URL: http://www.dsg.univr.it/documenti/OccorrenzaIns/matdid/matdid965805.pdf (siehe S. 2).

[2] Dr Bernard Wong und David Tein. „Critical Success Factors for ERP Projects“ (Jan. 2003), S. 1–2. URL: https://www.researchgate.net/publication/229022123_Critical_Success_Factors_for_ERP_Projects (siehe S. 2).

[3] Mark Brummel. *Learning Dynamics NAV Patterns. Create solutions that are easy to maintain, are quick to upgrade, and follow proven concepts and design.* Packt Publishing, 2015, S. 135–136. URL: https://www.microsoft.com/en-us/p/learning-dynamics-nav-patterns/fgqpf3h0qc1j?activetab=pivot%3aoverviewtab (siehe S. 2).

[4] Rabindra Sah. *Mastering Microsoft Dynamics NAV 2016*. Packt Publishing, 2016. URL: https://books.google.at/books?id=oLkrDwAAQBAJ (siehe S. 4).