Buzzwordbingo, auch Bullshitbingo genannt, meint: Inhaltslose Worthülsen sinnlos oder gar fehlerhaft in Werbe-Unsinn einzubinden, um eine nicht vorhandene Kompetenz vorzuspielen.
In inzwischen 4 Jahrzehnten habe ich so viele Hypes kommen sehen, erlebt und wieder gehen sehen. Und da ich seit 1993 mit Navision mein Geld verdiene, war insbesondere das „gehen sehen“ immer mit einem lachenden, statt einem weinenden Auge verbunden. Und so wird es nun Zeit, einmal einige der Hypes mit Blick auf „Mein“ Navision zu beleuchten.
Wichtig: Für jedes „buzzword“ gibt es sicherlich auch sinnvolle Anwendungen. Ich referiere hier ausschließlich auf die kaufmännische Welt, speziel von Dynamics Navision / Business Central.

Inhaltsverzeichnis

- Agile Entwicklung
- Continuous Delivery
- OOP – Object oriented programming
- In Memory Datenbank (Hana)
- DevOps
- Framework
- Low Code
- Cloud
- Edge Computing
- Micro services
- NoSQL
- Big Data
- KI Künstliche Intelligenz (Deep Learning)
- Blockchain
- GitHub
- Serverless Computing
- Chatbots
- Second Life
- Papierloses Büro
- SaaS – Software as a Service
- Scrum
- PbV – Pick-by-Voice
- Container, Docker, Kubernetes
- Disruption

- Mobile first
- Industrie 4.0
- KPI – Key Performance Indicators
- RAD – Rapid Application Development
- Extensions AL
- Kanban
- Kaizen

Agile Entwicklung

Agile Entwicklung war und ist in Navision & Business Central, zumindest in den Versionen von 1993 bis inklusive 2019 Spring Release, schon immer Standard! Lange bevor es diesen Begriff gab, war es gänzlich normal für Navision-Entwickler, direkt zum Mitarbeiter an den Arbeitsplatz zu gehen, sich das Problem am Bildschirm anzusehen und dann mal schnell eine Programmzeile umzustellen. Und dank der Navisionstruktur wurde die Programmänderung auch in wenigen Sekunden ausgeliefert.

Continuous Delivery

Wie bei Agile Entwicklung: Sekunden nachdem eine Programmänderung gemacht wurde, wurde diese auch vom System beim Anwender ausgeliefert. Musste früher notfalls mal ein Client neu gestartet werden, so schaffte RTC tatsächlich das Kunststück, diese Änderung sofort wirksam werden zu lassen.

OOP – Object oriented programming

Daten und Datenverarbeitungslogik bilden eine Einheit? 1993 hieß das noch „dbCallfieldcode“, 1996 wurde es geändert zu Record. Validate. Objekt-orientierte Programmierung in einer sauberen, verständlichen und nachvollziehbaren Art, ganz ohne Vererbung, Überladung und Parameterkonflikte. Type-Verträglichkeiten wurden bereits beim Speichern getestet und in vielen Fällen, wo auch immer möglich, vom System aufgelöst. Ansonsten gab es eine verständliche Klartextmeldung, keine Exception. Bis heute.

In Memory Datenbank (Hana)

Nach über 25 Jahren Entwicklung auf den verschiedenen Hardwaremodellen (Festplatten, SSD) & Navisionversionen (3.5x, Financials, Dynamics NAV, Business Central) & Serverversionen (Nativ, SQL) dieser Zeit: Ist die Hardware zu schwach, ist der Entwickler zu schlecht. Der native Server konnte leicht mit 1 Kern (mehr konnte er nicht nutzen) und 1

Gb Ram und 10 Festplatten (oder einer SSD) 100-120 aktive User verkraften. Allerdings habe ich in all der Zeit keinen Entwickler gefunden, der verstanden hat, warum die Transaktionszahlen mit der Anzahl der Festplatten skalieren. Oder der überhaupt wusste, was Transaktionszahlen sind…
Unter SQL sollten, auf einem sauberen System, 2 SQL Kerne pro 50 Mitarbeiter + 100 Mb pro Mitarbeiter ausreichen für ein immer performant reagierendes Datenbanksystem. Natürlich muss man auch hier die Transaktionszahlen im Auge haben, aber seit SSD's ist das ja auch für weniger bemittelte Hardwareeinrichter keine Frage mehr. Wobei ich es noch immer zu Hauf erlebe, dass jemand eine SQL-Datenbank und ein SQL-Transactionlog zusammen auf das gleiche Speichermedium (die gleiche Festplatte) legt. Teilweise durch die Hintertür eines Raid-5 Subsystems. Oder SQL-Dateien auf einem Memory-Pool. Da weiß man dann sofort: Gegen die Schleuse geschwommen und dann noch etwas nass und mit Kopfschmerzen die Konfiguration übernommen. Unter SQL kam die neue Herausforderung hinzu zu **wissen**, wie Schlüssel und Flowfields arbeiten. Und wie der SQL-Server mit Abfragen umgeht (Stichwort „**Result Set**„), auch das weiß bis heute kaum ein Entwickler. In Memory-Datenbanken sind Heilmittel, wenn die Entwickler unfähig sind, mit ihren Datenstrukturen umzugehen oder zum Testen grundsätzlich nur den Cronus-Mandanten nehmen

DevOps

Navision wird einmalig fertig installiert, egal, ob man die DOS-Version von 1993, die Windows-Native Version seit 1996 oder die RTC-Version seit 2012 benutzt. Somit endet die Ops-Seite von DevOps, denn ein Operator hat ab hier praktisch keine Aufgabe mehr zum stabilen Betrieb von Navision. Oft wird auch die Installation vom SQL-Server und des Navision-Dienstes direkt vom Developer mit erledigt. Weil er/sie am besten weiß (wissen sollte…), wie ein SQL-Server für Navision ticken muss. Die Entwicklung (Dev) von Navision erfolgt dann direkt innerhalb der Navision-Entwicklungsumgebung oder seit Extensions V2 (Business Central 2018) über Visual Studio Code. Auch hier hat der Operator, also der „Maschinist" im EDV Team, nichts mehr zu tun. Schon immer.

Framework

Natürlich ist Navision selber ein Framework, auch eines, welches sich weiteren Frameworks bedient, z.B. .Net, MFC. Aber damit hat ein Navision / Business Central Entwickler in der meisten Zeit keinen Kontakt: er konzentriert sich auf Datenmodelle (Tabellen), ihre Beziehungen untereinander (Relations), und auf die Geschäftslogik (C/Side, C/AL, AL, alles irgendwie das gleiche). AngularJS, Drupal, Ruby, JavaScript, VCL, Application Frameworks, Domain Frameworks, Class Frameworks, Komponenten-Frameworks? Ja, all das ist auch

irgendwie Navision. Nur: Den Programmierer / Entwickler interessiert das alles nicht. Leider treibt Microsoft inzwischen selber alle paar Monate eine neue Sau durch die Navision-Dörfer. Und kopiert damit die ständige Hektik der „realen Welt“ da draußen in die Navision-Welt hinein. Doch wenn man sich als Navi-Entwickler mal in Ruhe zurücklehnt, dann ist das Ganze am Ende doch wieder alles irgendwie das gleiche, auch AL mit VS Code. Und so entsteht dann wieder die nötige Ruhe für gute Lösungen.
Das Navision / Business Central „Framework“ zeigt sich -vor allem- durch die -als Genial zu bezeichnende- Entwicklungsumgebung, die alle in anderen Programmiersprachen üblichen Fallstricke (Racetime-conditions, garbage collection, Managed Code, Transaction & commitment…) vom Entwickler fernhält. Seit dem RTC gibt sich Microsoft redlich Mühe, wieder diverse Probleme dieser Zeit (plötzliche Systemstillstände, schreckliches Stringhandling, Sort-On-Demand…) in die vormals glückliche Navision-Welt einzuführen. Wenn man die Fallstricke aber kennt, kann man sie erstens schnell erkennen und zweitens schnell umgehen.

Low Code

Wenig Code schreiben, um eine Applikation zu entwickeln? Navision hat’s erfunden. Schon im letzten Jahrtausend! Für Rapid Prototyping inklusive Mockups und Dokuscreens konnten bis zur 2009R2 (ohne RTC) komplett funktionierende Applikationen inklusive Reporting, Sortierungen, Suchen & Filtern gänzlich ohne eine Zeile Code entstehen. Das geht heute leider nicht mehr. Aber auch mit AL immer noch schneller als in jeder anderen mehr oder weniger aktuellen Entwicklungsumgebung.

Cloud

AWS, Azure, Private Cloud… „Es gibt keine Cloud. Es gibt nur Computer, die jemand anderem gehören“. Und dafür bezahlt man. Denn der Computerbesitzer will seine Computer auch bezahlt bekommen. So einfach funktioniert dieses Geschäft. Mit dem Unterschied, dass man nun ein hunderte Kilometer langes Kabel zwischen seinem Monitor und diesem fremden Computer hat. Und dass man nicht weiß, wo dieses Kabel endet; wo also Ihre Daten sind. Und man weiß auch nicht, wer alles am anderen Ende des Kabels sitzt und dass dieses auch mal getrennt werden kann. Und der Computer kann auch schon mal abgeschaltet werden. Nein? Wo sind die Windows-Phones? Was passierte mit der deutschen Microsoft-Cloud? Was wird aus Google Cloud Print, google wave, Google+? Was wurde aus all den Blackberry-Services? Was ist mit Irista von Canon? Was passierte mit Lima? Wie speichern Sie heute Ihre Daten vom Robin-Smartphone?
Auch bei noch laufenden Clouddiensten stellt sich, gerade in Deutschland, die Frage: Reicht meine Bandbreite, um den Dienst in meiner Firma überhaupt betreiben zu können? Was,

wenn ich meine Firma aus Berlin Zentrum nach Brandenburg umziehe? Einen eigenen Server kann ich mitnehmen, einen Glasfaseranschluss nicht.
Die homöopathischen Installationen für ein Business Central auf der Azure-Cloud sprechen für sich. Es wird noch nicht einmal berücksichtigt, welche zusätzlichen Einschränkungen man sich mit dieser Entscheidung einhandelt. Man sollte, meiner Meinung nach, nichts in „die Cloud“ legen, auf das man nicht mindestens einen Tag verzichten kann. Und Kosten sparen? Wie bemerkt: Es ist ein Server, der jemand anderem gehört. Und der andere stellt diesen Server nicht zur Verfügung, weil er so ein toller Menschenfreund ist. Aber natürlich gibt es, gerade für Software, Mietmodelle, die bewusst so kalkuliert sind, dass dem Kunden die monatliche Gebühr statt der Anschaffung schmackhaft gemacht wird.

Edge Computing

Wie schon beim Cloud computing festgestellt: Nicht jede am Standort erreichbare Bandbreite genügt auch den benötigen Anforderungen. Die Lösung? Edge-Computing! Im einfachsten, der Definition am nächsten kommend, wandert so der gewünschten Rechendienst aus dem eigenen gesicherten Serverraum hin zum per Vierkantschlüssel gesichertem Plastikschrank an der Straßenkreuzung… eben Edge-Computing… Computing an der Ecke (oder Schnittstelle) zwischen eigenem Netz und fernem Rechenzentrum.

Micro services

Kleiner Service. Unabhängig, einfach zu warten, einfach zu skalieren. Sehr praktisch, um einen Wechselkurs im Internet abzufragen. Was aber, wenn zu dem abgefragten Wechselkurs auch gleich eine Transaktion durchgeführt werden soll, z.B. ein Währungstransfer? Dann wird aus den mindestes zwei oder noch mehr nötigen Microservices schnell ein Macroservice… Ein nicht handelbarer Macroservice. Nicht umsonst werden Flugpläne und -buchungen, Stromlieferungen, Rentenversicherungen noch immer größtenteils auf Mainframes abgebildet. Weil dort eine Transaktion eben auch garantiert als Ganzes abgeschlossen und bestätigt werden kann. Ein ERP mit tausenden von Microservices? Konsistente Preisermittlung zusammen mit einem garantiert vollständig abarbeitbaren Auftrag bei Berücksichtigung eines Kreditlimits? In Navision startet man eine Transaktion und bucht in einem Rutsch Bestände, Salden, Kostenrechnung, Fertigungsaufträge und Verkäuferumsätze. Und bei einem Abbruch stehen wieder alle Daten konsistent im Ursprungszustand zur Verfügung. Schlicht unmöglich mit Microservices. Es gibt sicherlich auch sinnvolle Anwendungen für diese. Aber nicht in der ERP-Welt.

NoSQL

Natürlich ist es schön, auch gänzlich unstrukturierte Daten in einer Art Datenbank abzulegen. Früher hieß das Dateisystem oder auch Lotus notes. Insofern auch nur alter Wein in neuen Schläuchen. Nur was macht man dann mit den unstrukturierten Daten? Man bildet Metadaten, Attribute… klassische SQL-Strukturen. Und Umsätze, offene Posten, Lagerbestände und Sachkontensalden benötigen schlicht kein No-SQL… eben: No NoSql. Für ein kaufmännisches System sind strukturierte Daten wie sie in einer dBase-, DB2-, Firebase-, oder eben Navision bzw. SQL Datenbank vorliegen, das tägliche Brot. Man braucht nicht weniger. Aber auch nicht mehr.

Big Data

Das Schlagwort von 2018-2019. Spricht heute noch jemand davon? Es gibt viele Informationen die man erst in einer großen Masse von Daten findet. Schaltzyklen von Verkehrsampeln anhand von Verkehrsflüssen. Defekterkennung durch Laufzeit- oder Temperaturerkennung. Verbreitungsmuster von Epidemien. In aller Regel bedeutet „Big Data“ aber, genauso wie bei KI, dass nur jemand zu faul war, sich selbst an die Mustererkennung zu machen. Ampelschaltzyklen könnte ich auch direkt an den Ampelrechnern abgreifen. Wenn man aber lieber kostenlos abzufragende Handys statt teurer Verkabelung von Ampelrechnern nutzen will, dann ist Big Data wieder ein tolles Mittel für nachgeschaltete Mustererkennung, was uns zur KI bringt.
Heute ist das *neue* Big Data dann doch wieder **Smart Data**: Intelligent aufbereitete, kompakt erfasste und leicht zu verarbeitende Daten. Und BigData braucht man nur noch als Trainingsmaterial für die berühmte KI. Danach sind die Daten nur noch Ballast.

KI Künstliche Intelligenz (Deep Learning)

Künstliche Intelligenz? Gibt es nicht. Gab es nicht in den 60ern, als Eliza noch eine Schlagzeile wert war. Gab es auch nicht später, bis heute nicht. Was uns heute als KI verkauft wird, ist eine automatisierte Mustererkennung aus Massen von vorindizierten Daten, eben wieder Big Data. Und je besser man diese Mustererkennung wieder versteht, desto ernüchternder zeigt sie sich: Mhm… nicht ganz! In der geradezu magisch erscheinenden Bilderkennung werden Züge nicht etwa an der Lock, sondern an den Gleisen erkannt. Pferde werden nicht an der Kopfform, sondern an Copyrighthinweisen erkannt. Daher werden auch z.B. Gesichterkennungen (Nicht „Gesichtererkennung“) durch Vektorenabgleiche durchgeüfhrt. Hochkomplexe Programmierung. Aber eben auch effektiv, im Gegensatz zu der oft eher „präzise ratenden KI“. In Navision bzw. genauer Business Central kann man dies wunderbar an der Zahlungsmoral von Debitoren (Kunden) erkennen.

In fast jeder Navision-Installation, die ich betreute, gab es in den Debitoren-Rechnungsposten die Spalte „Überzug in Tagen“. Mit einer Zeile im Programmcode beim OP-Ausgleich wurde dort ein Überziehen der Fälligkeit protokoliert. Und so konnte sehr simpel und sehr genau ermittelt werden, ob ein Kunde ein zuverlässiger Zahler ist und mit welchen Ø Überziehungen i.d.R. pro Kunde gerechnet werden muss. Simpel und genau. In Business Central gibt es nun eine KI-Lösung, die mit viel Gedöns und viel Einrichtung zu einem offenen Posten ausgegeben kann „Wird vielleicht überzogen“. Echt! Mehr nicht! Im kaufmänischen Sektor, bei Beständen und Salden, bei Überweisungen und Mahnungen brauche ich schlicht keine KI. Ich brauche vernünftige Buchhalterkenntnisse, gute Verkäufer- und Einkauferkenntnisse, solide Warenwirtschaftskenntnisse. Aber Navision muss ja auch keine selbstfahrenden Autos vor dem Zusammenstoß mit Zügen bewahren…

Blockchain

Was sollte nicht alles einfacher, günstiger und sicherer werden mit der Blockchain. Arztberichte, Containerverfolgung, Auflösung der Banken in wenigen Monaten, Demokratisierung des Geldes. Selbsterfüllende Verträge. Ein sehr hoher Anspruch für eine dezentrale, replizierende Datenbank – mehr ist die Blockchain, bei aller Ehre, erst einmal nicht.
So etwas „ähnliches“ gibt es im Navision-Büro auch: Dort nennt es sich Kontoauszug, welcher die Zahlungen und Geldeingänge synchronisiert. Und das seit vielen Jahren, auf Wunsch auch automatisch.
Was blieb bisher von der Blockchain? Eine ganz gute Methode, um illegale Geschäfte abzuwickeln oder Computererpressungen zu bezahlen. Dumme Menschen mit irren Versprechungen zu inhaltslosen Geldzahlungen zu bewegen. In der echten Wirtschaft dann doch eher eine günstige Werbemöglichkeit. Neue Schlagzeilen á la „Erster Gast hat mit Bitcoin bezahlt“ erinnert doch sehr an die erste Eisenbahnfahrt von Nürnberg nach Fürth. Bitcoin bleibt spannend, Blockchain wird seinen Markt finden. Doch: Ein durch Marketing-Hypes getriebenes „Alles muss in die Blockchain“ ist etwa genauso doof wie ein unüberlegt dahin geplappertes „Alles muss in die Cloud“.

GitHub

Eine Verwaltung, in der ich verschiedene Versionen eines Programms ablegen kann. Das hieß früher Texteditor oder Sicherheitskopie und liegt heute stylish in der Cloud. OK, Merging, Branching and Forking sind neu… Und irgendwie doch totaler *Overhead*, wenn man eh nur wenige Zeilen ändern muss. Das Original hat man ja in dem betriebsbereiten Backup von gestern… oder etwa nicht?

Serverless Computing

Als Begriff selber eines der dümmsten Buzzwords, das man sich ausdenken kann. Gemeint ist aber, dass man auf einer abstrahierten Plattform selbst Funktionen laufen lassen kann, ohne sich um die dafür notwendigen Rahmenbedingungen (Serverhardware, Leistung, Konfiguration, oft auch Sicherung, etc.) Gedanken machen zu müssen. Natürlich muss das „Serverless“ zur Verfügung gestellte Framework auch 100% zur Aufgabenstellung passen. Das sollte man besser vorher prüfen.
Unter Navision benutze ich dieses Buzzword manchmal aus Spaß, um damit zu sagen, dass die Navision-Umgebung (Server, Client oder Service) nach der korrekten Einrichtung „einfach vergessen werden kann“. Wenn man sauber programmiert, stimmt das auch. Ich hatte Kunden, die nicht wussten, wo ihr nativer Navision-Datenbankserver stand, weil sich über Jahre niemand um diese Kiste gekümmert hat.

Chatbots

Eine Plage. Insbesondere weil man mangelhaft (=billig) geschulte Supportmitarbeiter oft nicht von schlecht programmierten Chatbots unterscheiden kann. Und doch waren die Chatbots vor ca. einem Jahrzehnt DER Hype, den jede Firma brauchte. Und heute nerven die damals schludrig hinprogrammierten Systeme manchmal noch auf Webseiten, weil jemand vergessen hat, dass da noch das Script von damals läuft. Einen Kultstatus hat dabei „Eve“ erlangt, eine kleine animierte Figur auf Yellow-Strom, die sich -bei Eingabe der richtigen Begriffe- nackig gemacht hat. Sie wurde leider nicht „vergessen“, sondern irgendwann echt abgeschaltet. Schade.

Second Life

Linden-Dollar… erinnert sich noch jemand an dieses neue „must have“ von 2003? Webshops, gerade erst für viel Geld aus der Taufe gehoben, werden unnötig. Telefonate sterben aus. Jeder Support und jedes Marketing musste unbedingt dort präsent sein… Und heute? Kommunikation findet dank Browser, Google und SEO-Optimierung zum allergrößten Teil über Texte statt, wie im 18. Jahrhundert. Diese werden tatsächlich auch noch gelesen und nicht per Micro-Chip-Implantat direkt ins Hirn übertragen… Leider! Leider?

Papierloses Büro

Endlich mal ein deutsches Buzzword Und auch schon fast so alt wie ich. Und genauso alt wie der Begriff „Papierloses Büro“ ist auch die widersprechende Studie zum

Papierverbrauch selbst. Allerdings erkenne ich seit 2020 den wer-weiß-wie-vielten Silberstreif am Horizont! „Dank“ Corona -und dem damit oft nötig gewordenen Home-Office (noch so ein Buzzword, aber diesmal ein richtig gutes, wie ich finde…)- versuchen viele Firmen nun wirklich konsequent, zumindest ihre Eingangs- und Ausgangsrechnungen auf elektronische, papierlose Belege (PDF etc.) umzustellen. Und das, dank gelockerter staatlicher Regeln, auch immer erfolgreicher. Wenn alle an einem Strang ziehen…
Gerne zeige ich Ihnen auch mal meine Version einer papierarmen Eingangsrechnungverarbeitung.
Und doch… Wenn ich Buchungsfehler in der Finanzbuchhaltung, Warenwirtschaft oder Anlagenbuchhaltung suche, dann mache ich selbst doch immer zuerst einen Papierausdruck vom entsprechenden Konto…

SaaS – Software as a Service

Vom Benutzer maximal parametrisierbare, aber sonst nicht ernsthaft veränderbare Programme werden komplett ohne lokale Installation über das Internet genutzt. Google Mail, Microsoft Office 365, Adobe… Tolle Lösungen, kinderleicht zu nutzen, auf jedem Gerät verfügbar. Inzwischen auch Business Central! Und die Regel lautet: Nutzen Sie es genauso, wie wir uns das vorgestellt haben… auch wenn wir überhaupt keine Ahnung von Ihren Geschäftsprozessen, Ihren Sonderpreisen und Ihrer Vertreterabrechnung haben. Na ja, ganz so extrem ist es bei BC dann doch nicht. Über die Extensions kann man komplette Lösungen und Modifikationen in BC integrieren. Man muss halt nur mit der Gewissheit leben, dass Microsoft da mal eben ein Update drüber spielt und Ihre Warenwirtschaft & Finanzbuchhaltung ein paar Tage nicht funktioniert, weil ein „Plugin“ (so kann man die Extensions durchaus verstehen) nicht so ganz zu der Änderung passt. Aber für einfache Firmen ist BC als SaaS durchaus eine Option. Ich kenne nur keine so einfach gestrickte Firma… Ach ja… Erinnert sich noch jemand an die Saas der 70er? Nein? Noch heute laufen alle Flug- und Zugbuchen (weltweit?) über diese lustigen grünen Terminals… mit serieller Anbindung an einen Großrechner. Energiebroking, Bankenbewegungen, Aktienhandel, Flugbuchungen, Containertracking: Großrechnersysteme, die es per serieller Anbindung erlauben, von überall auf der Welt angesprochen zu werden. Der Großrechner ist tot… es lebe der Mainframe

Scrum

Kennt eigentlich jemand den Wort-Ursprung? Scrum ist das Gedränge im Rugby bei intensiven Ballkontakten. Gemeint ist damit, dass kleine Entwicklungsteams nur noch ein Ziel vorgegeben bekommen. Den Weg dahin sollen Sie selbstständig festlegen, in der Hoffnung, dass sie ohne Regulierung schneller ihr Ziel erreichen. Insofern könnte man auch

das Gedränge in einem Ameisenhaufen als selbstorganisierendes Team bezeichnen. In Navision / Business Central ist diese Methodik weniger verbreitet, da oft einzelne Entwickler statt Entwicklerteams komplette Lösungen von der Aufgabenstellung bis zum Rollout betreiben (project owner). Aber natürlich ist SCRUM auch unter / mit Navision / BC abbildbar, bedingt durch die rasend schnellen Ergebnisse vielleicht sogar noch besser als in anderen Umgebungen.

PbV – Pick-by-Voice

Ein Logistikhype, den ich persönlich nie verstanden habe… genauso wie viele Anwender die Stimmen nicht verstanden haben und der nachgeschaltete Computer dann wieder nicht die Rückfragen der Picker verstanden hat. Ich kenne nicht viele Firmen, die PbV eingesetzt haben.. Und ich kenne **keine** Firma, die PbV **erneut** einsetzen würde. Vermutlich wird es den einen oder anderen Leser geben, der PbV liebt und verteidigt… die meisten werden nur, wie ich, mit der Schulter zucken.

Container, Docker, Kubernetes

Virtuelle Maschinen, Container… Werkzeuge, um teure Hardware besser auszulasten, aber kein Allheilmittel für irgendetwas… und schon gar nicht ein Wundermittel. Container können, je nach Einsatzzweck, durchaus auch Nachteile gegenüber virtuellen Maschinen haben. Und beide können auch mal ein Rennen gegen dediziertes Blech verlieren. Recht wenig hilfreich ist dann noch besonders die Anstrengung von Microsoft, die Studio Code Entwicklungsumgebung für Business Central/AL in vielen (allen?) Anleitungen als Docker-Image zu bewerben. Eine gänzlich unnötige Hürde, die schon viele Interessierte von der Extensions-Entwicklung abgehalten hat. Und **die** ist echt geil!

Disruption

Was sich nicht täglich selber neu erfindet, wird sich selbst nicht überleben… So oder so ähnlich lautet das Damoklesschwert über allen Programmen, die nicht jeden Monat mit der allerneusten Technologie neu compiliert werden. Schade, dass Millionen Zeilen von ABAS, Navision (C/)AL, RPG und Cobol nichts davon mitbekommen haben. Nicht jede Web-Ratespielprogrammierung für einen Radiosender, programmiert auf dem neuesten Software Paradigma, ist von solch wirtschaftlicher Bedeutung, dass sie in 10 Jahren noch gepflegt werden wird. Weltweite Flugbuchungen in Cobol hingegen schon.

Mobile first

Hauptsächlich von Google geprägt. Immer mehr Anwender wandern zum Handy ab. Sicherlich so korrekt. Aber… Kennen Sie einen Einkäufer, der den Stahlbedarf der nächsten drei Wochen auf seinem Handy kalkuliert? Können Sie sich einen Buchhalter vorstellen, der 1.200 OP's am Tag am Handy überprüft und ausbucht? Amazon, Whatsapp und Facebook sind nicht immer die richtigen Maßstäbe fürs Business.

Industrie 4.0

Vielleicht das **Buzzword** 2.0, der **Hohlsprecher** 1.0. Vernetzte Maschinen? Rückmeldungen von Produktionsdaten? Roboter, die selbst ihre Werkzeuge wechseln können? Das ist schon lange Standard in der Produktion, wo es Sinn macht. Eine Maschine, die 24/7 7365 Pommesgabeln produziert, muss nicht unbedingt mit der Losgröße 1 umgehen können…

KPI – Key Performance Indicators

Ja… WAS sind den Key Performance Indicators? Umsatz pro Mitarbeiter? Kosten pro Tag? Bestandsveränderung im Vergleich zum Vorjahr? Alles wichtige Analysezahlen, die aber auch schon ohne diesen Begriff in 100 Jahre alten Büchern für die Ausbildung zum -damals so genannten- Kaufmannsgehilfen zu finden waren.

RAD – Rapid Application Development

…Schneller als mit Navsion / Business Central können Sie kaufmännische Lösungen nicht entwickeln. Ich habe zumindest in 30 Jahren „Revolutionäre Versprechen" nichts gefunden, das dies ermöglichte. Sonst hätte ich mich sicherlich bereits umentschieden.

Extensions AL

Teilen wir das auf: Extensions V2 (!) an sich sind der größte Schritt, den Navision / Business Central jemals gemacht hat. Sauber (!) programmiert, erlauben sie echt die Erweiterung von Navision (aus Anwendersicht), so einfach wie eine App auf dem Handy. Sie ergeben zwar erst einmal auch die gleichen Probleme wie Apps, z.B Abhängigkeiten von bestimmten Navision-Versionen, Probleme beim Update der Basisapplikation etc.
Aber das liegt in der Natur der Sache und ist bei „Nativer" Programmierung noch viel schlimmer. Und daher nicht als Nachteil zu bewerten.
Allen Beteuerungen zum Trotz: Extensions selbst sind kein Grund gewesen, die integrierte

Entwicklungsumgebung aufzugeben. Das ist eine reine politische Entscheidung und da hat Microsoft ja nicht immer den Kundengeschmack getroffen… Insofern könnte man Windows ME, Ribbons, Windows 8 auch ganz gut zu den Buzzwords zählen.

AL über Visual Studio Code ist, aus Entwicklersicht, zumindest aus meiner, hingegen ein riesiger Rückschritt. AL wird nun als „echte Programmiersprache“ oder „Navision wird erwachsen“ beworben. Es ist weiter keine echte Programmiersprache, zumindest nicht im Vergleich zu C#, Visual Basic, C+, Delphi, etc..
Und das ist auch vernünftig und gut so!
In einer „echten“ Programmiersprache würde ich mit so etwas wie #include beliebige Erweiterungen zu meiner Sprache hinzuladen. Microsoft hat aus AL sogar Net verbannt, zumindest aus der Azure, und für alle folgenden Versionen angekündigt.
Umgekehrt würde ich mir in einer „echten“ Programmiersprache alleine für eine Transaktions-Bündelung schon die Finger verbiegen.
AL ist genau das, was C/Side bzw. C/AL (Unter Navision 3.5x hieß das auch schon „AL“!) schon immer war: Eine hochperformante, optimal auf kaufmännische Prozesse fokussierte Entwicklungsumgebung. Nicht mehr, aber auf jeden Fall auch nicht weniger. Nur ist die Entwicklung nun ungleich komplizierter geworden, verglichen mit der bisherigen RTC-Umgebung. Und sowohl RDLC wie auch Studio Code sind riesige Rückschritte gegenüber der vollständig integrierten Umgebung inkl. Reportdesigner von Navision bis 2009R2. Ein Argument bei diesem Vergleich war z.B. mal, dass man im RDLC jetzt auch um 90° gedrehte Texte, z.B. als Firmenwerbung, auf dem Ausdruck platzieren kann. Das habe ich von 1993 bis 2015 nicht vermisst und bis heute nicht einmal gebraucht. Und auch das Hyperlinking hätte man, wenn man wollte, in den Reportdesigner einbauen können. Hat das mal jemand echt aktiv benutzt? Ich meine, nicht nur zu Show-Zwecken?
OK, der Fairness halber: Seite 1 von x und farbiger Druck, z.B. Rot für negative Salden, ist echt cool und ein echter Gewinn. Aber auch das ginge mit dem altem Reportdesigner, wenn Microsoft nur gewollt hätte. Und wenn ich tauschen könnte, dann würde ich lieber auf rote Salden als auf den nativen Reportdesigner verzichten

Ich lasse mich zu jeder Tages- und Nachtzeit auf folgenden Vergleich / Wettbewerb ein, in der ich meine Aussage beweise: Wir definieren eine typische kaufmännische, datengetriebene Aufgabenstellung und starten mit einer jeweils passenden Navision-Setup-DVD (gerne als Image) auf einem blanken Windows-Rechner. Mein Einsatz: ein gehobenes Abendessen in einem gehobenen Restaurant, der Langsamere zahlt.

Und das letzte Argument für AL, dass jetzt „Jeder, der Programmieren kann, auch Navision programmieren kann“: Entschuldigung, aber das ist so unglaublich doof… Das ist nicht nur

doof, das ist auch noch extrem gefährlich! Natürlich konnte auch schon bisher jeder, der den Designer aufrufen konnte, Navision auch kaputt machen. Aber es gab einen natürlichen Respekt vor der Umgebung: Erst Business Central verstehen, dann rangehen und ändern. Das hielt immerhin einige Vollpfosten davon ab, Navision kaputt zu programmieren. Die anderen fand man dann bei Mibuso. Jetzt bewirbt Microsoft das ganz offen, dass nun jeder in dieser Warenwirtschaft programmieren könne. Auch ganz ohne Warenwirtschaft- oder Buchhaltungskenntnisse. Ich freue mich drauf… So oder so.
Nicht umsonst rät sogar Thomas Heilsberg, Bruder des Erstellers von Turbo Pascal und heute immer noch ein entscheidender Architekt von Navision / Business Central: To be an excellent Microsoft Dynamics NAV Developer, understanding business processes is maybe even more crucial than understanding the language, objects, and design patterns. Frei übersetzt: Um ein guter Navision-Programmierer zu sein, ist das Verstehen der Geschäftsprozesse vermutlich wesentlich wichtiger als das Verstehen der Entwicklungsumgebung.
Ach ja… mit Extensions hat Navision endlich technisch zu SAP aufgeschlossen… Und auch die gleichen Probleme wie Lesbarkeit und Fehlersuche mitgeerbt. Hätte man 30 Jahre später besser machen können…

Werbung: In diesem Buch findet man diese Aussage und noch viel mehr zum Verstehen von Navision:

Kanban

„Karte“. Eine Bedarfgesteuerte („Pull“) Nachschubsteuerung. Findet in der westlichen Industrie leider nur ein Nischendasein. Aber wenn Sie es mal für Ihre Kunden einsetzen wollen?

Kaizen

Kaizen ist eine in Japan entstandene Philosophie der ewigen Veränderung. In der Industrie ist Kaizen insbesondere durch die Umwälzungen und Verbesserungen bei Toyota bekannt geworden.
In seiner industriellen Geschichte besteht Kaizen vor allem aus den drei zu vermeidenden Mu:
Muda Verschwendung. Dinge, Arbeitszeiten, Materialien sollen nicht verschwendet werden. In der westlichen Interpretation von Kaizen ist vor allem das Muda angekommen.
Mura Abweichungen in den Prozessen. Prozesse sollen normiert und standardisiert werden

und danach in dieser Form gelebt werden. Wer schon einmal bei einer ERP-Planung und -Einführung dabei war, weiß: Die Zeit und das Geld gehen nicht für die 90% der einfachen Prozesse drauf. Fehler entstehen und Zeit wird verbrannt, weil sich praktisch jede Diskussion um Features für die 10% (oder weniger) der Abweichungen dreht.
Muri Überlastung von Mitarbeitern und Maschinen. Dieses Mu wird praktisch immer „vergessen“. Gerade in der westlichen Industrie gibt es das schädliche Mantra der „Nahe 100% Auslastung“ von Mitarbeitern und Maschinen. Dies ist ungesund bis tödlich. Die Ergebnisse erlebt jeder von uns (fast) jeden Tag: Staus auf den Autobahnen, Verspätungen und Zugausfälle bei der Bahn und bei Flügen. Warten auf öffnende Kassen im Supermarkt, lange Wartezeiten in der Telekom-Hotline. Auslastungen bis höchstens 85 % halten Prozessabläufe geschmeidig. Die Leerlaufzeiten sparen Geld!