
PŘÍRUČKA · 2026

Business Central *v době AI*

Příručka pro vývojáře a konzultanty

47 kapitol · 6 modulů · kompletní AL kód

Prompt engineering · Claude Code · GitHub Copilot

Azure OpenAI · RAG · AI agenti · MCP servery

AUTOR

Miloš Mikulášek

Business Central v době AI

Příručka pro vývojáře a konzultanty

Miloš Mikulášek

Vydání 2026 · Verze 1.0

Impressum

Business Central v době AI *Příručka pro vývojáře a konzultanty*

© 2026 Miloš Mikulášek. Všechna práva vyhrazena.

Žádná část této publikace nesmí být reprodukována, uložena ve vyhledávacím systému nebo přenášena v jakékoli formě nebo jakýmkoli prostředky — elektronickými, mechanickými, fotokopírováním, nahráváním nebo jinak — bez předchozího písemného souhlasu autora.

Ukázky kódu obsažené v této knize jsou dostupné na adrese github.com/ai4bc-dev a mohou být volně použity ve vašich projektech.

Autor	Miloš Mikulášek
Web	ai4bc.dev
Email	milos.mikulasek@ai4bc.dev
GitHub	github.com/ai4bc-dev
Verze	1.0
Vydáno	2026
Jazyk	Čeština

Názvy produktů a společností zmíněné v této knize mohou být ochrannými známkami jejich příslušných vlastníků. Microsoft, Business Central, Azure, GitHub Copilot a Dynamics 365 jsou registrované ochranné známky společnosti Microsoft Corporation. Claude je ochranná známka společnosti Anthropic, PBC.

Předmluva

Tato kniha vznikla z jednoduché frustrace: příliš mnoho schopných Business Central specialistů tráví příliš mnoho času na věcech, které by za ně mohla dělat umělá inteligence.

Viděl jsem BC vývojáře, kteří stráví hodiny hledáním správné AL syntaxe pro věc, kterou by Claude Code napsal za třicet vteřin. Konzultanty, kteří ručně prepisují zápisky ze schůzky do funkční specifikace, pře-

stože by AI zvládla první draft za dvě minuty. Freelancery, kteří odmítají projekty, protože na dokumentaci nemají čas — přestože AI by jim ten čas dala zpět.

Zároveň jsem viděl druhou skupinu lidí. Ti, kteří AI přijali a integrovali do svého pracovního postupu. Na rutinních úkolech píše kód 3–5× rychleji. Specifikace, které dřív trvaly den, teď trvají hodinu. Zákazníkům nabízejí věci, které jejich konkurenti ještě ani nevidí.

Tato kniha je pro první skupinu, která se chce stát druhou.

Pro koho je tato kniha

Tato kniha je určena třem skupinám čtenářů, kteří se denně setkávají s Business Central:

BC vývojáři a programátoři znající AL jazyk, kteří chtějí systematicky integrovat AI do svého vývojového procesu. Naučíš se pracovat s Claude Code a GitHub Copilotem, volat Azure OpenAI API přímo z AL kódu a stavět vlastní AI funkce pro zákazníky.

BC konzultanti a funkční analytici, kteří nepíší kód nebo píše jen minimálně. Naučíš se používat AI pro tvorbu funkčních specifikací, testovacích scénářů, uživatelské dokumentace a zákaznické komunikace — a ušetříš hodiny každý týden.

Freelance BC specialisté, kteří pracují samostatně nebo v malých týmech. Naučíš se, jak AI funguje jako „druhý vývojář“ — pomáhá s kódem, dokumentací i komunikací a umožňuje ti konkurovat větším firmám.

Kniha předpokládá základní znalost Business Central a jeho ekosystému. Nemusíš být expert na AL programování, ale měl bys rozumět tomu, co BC dělá a jak funguje.

Co tato kniha nepokrývá

Tato kniha není úvodem do Business Central samotného. Nepokrývá instalaci BC, základy účetnictví, ani implementaci standardních BC procesů. Předpokládáme, že Business Central znáš — a učíme tě, jak do něj přidat umělou inteligenci.

Kniha také není akademickým pojednáním o teorii strojového učení. Nezajímá nás, jak přesně fungují neuronové sítě uvnitř GPT-4o. Zajímá nás, jak tyto nástroje prakticky použít, aby nám ušetřily čas a přidaly hodnotu zákazníkům.

Předpoklady

Pro maximální přínos z této knihy doporučujeme:

- Alespoň základní znalost Business Central (orientace v systému, znalost hlavních modulů)

- Pro kapitoly o AL programování: zkušenost s AL jazykem nebo jiným programovacím jazykem
- Přístup k Microsoft 365 nebo Azure předplatnému (pro praktické příklady v Modulu 3)
- Nainstalovaný VS Code s AL Language Extension (pro Moduly 1 a 3)

Modul 0 (Úvod a orientace), Modul 2 (AI pro konzultanty) a Modul 5 (Akční plán) jsou dostupné komukoli bez programátorských předpokladů.

Jak je tato kniha organizována

Kniha je rozdělena do šesti modulů (0–5), které na sebe logicky navazují — ale každý modul je zároveň čitelný samostatně.

Modul 0: Úvod a orientace (3 kapitoly) vysvětluje, proč AI mění ERP svět, jaké nástroje existují a jaká jsou realistická očekávání. Pokud s AI teprve začínáš, začni zde.

Modul 1: AI jako vývojářský parták (11 kapitol) se věnuje každodenním nástrojům pro vývojáře — prompt engineering, Claude Code, CLAUDE.md, GitHub Copilot, AI pro testování a workshop, kde postavíme kompletní AL extension od požadavků po hotový kód.

Modul 2: AI pro denní práci konzultanta (9 kapitol) pokrývá použití AI pro konzultantský a analytický den: specifikace, testovací scénáře, dokumentace, zákaznická komunikace, troubleshooting, SQL analýza a migrace dat.

Modul 3: AI integrace přímo v BC (10 kapitol) je technické jádro knihy. Naučíš se konfigurovat Microsoft Copilot v BC, volat Azure OpenAI z AL kódu přes HttpClient a postavíš tři praktické demo integrace — kategorizaci položek, AI vyhledávání a inteligentní návrhy. Projdeme také Power Platform AI Builder, Copilot Studio, GDPR aspekty a jak prodat AI řešení zákazníkovi.

Modul 4: Pokročilé techniky (6 kapitol) pokrývá RAG architekturu s kompletním Python kódem, AI agenty s Azure Function implementací, rozhodovací framework fine-tuning vs. prompting, MCP servery s plnou TypeScript implementací, emerging nástroje a kariérní roadmapu.

Modul 5: Akční plán (4 kapitoly) uzavírá knihu konkrétními kroky: 30denní implementační plán s týdenními checklisty, jak přesvědčit firmu s ROI kalkulací, kde se dál vzdělávat bez FOMO a závěrečné povzbuzení.

Přílohy obsahují kompletní referenci AL kódu ze všech kapitol (Příloha A), knihovnu 12 prompt šablon pro běžné BC úkoly (Příloha B) a 13 praktických cvičení ke všem modulům (Příloha C).

Konvence použité v této knize

V celé knize používáme jednotné typografické a textové konvence.

Kód inline označuje AL kód, příkazy terminálu, názvy procedur, proměnných, tabulek a polí v textu.

```
// Bloky kódu vypadají takto
procedure MujPriklad(): Text
begin
    exit('Funkční AL kód, ne pseudokód');
end;
```

Všechny ukázky kódu jsou funkční a testovatelné v Business Central verze 27 a novější. Namespace ukázkového kódu je **BCDemoAI**, komentáře jsou česky.

- **Poznámka** označuje doplňující informaci, která se hodí znát, ale není nezbytná pro pochopení hlavního textu.
- **Upozornění** označuje bezpečnostní nebo GDPR aspekt, který je důležité respektovat v produkčních nasazeních.
- **Tip** označuje praktickou zkratku nebo osvědčenou praxi z reálných projektů.

Zkratky a pojmy z angličtiny: v oboru BC a AI se používá velké množství anglických termínů, které nemají přirozené české ekvivalenty (codeunit, prompt, deployment, embedding). Tyto termíny ponecháváme v anglické podobě a zavádíme je při prvním výskytu.

Každá kapitola obsahuje na konci sekci **Vyzkoušejte si** s 2–3 praktickými cvičeními navázanými na obsah kapitoly. Vybraným kapitolám předchází nebo je doplňuje **Case study** — anonymizovaný příběh z reálné BC praxe.

Ukázky kódu a materiály

Veškerý kód z této knihy je dostupný na github.com/ai4bc-dev. Repozitář obsahuje:

- Kompletní demo AL extension (Purchase Approval s AI notifikacemi)
- Azure OpenAI Helper codeunit (BC v27, plně funkční)
- Python RAG indexer pro BC data (Azure AI Search)
- TypeScript MCP server pro BC OData API
- Python Azure Function pro email-to-order AI agenta
- Sadu prompt šablon pro vývojáře a konzultanty (12+ šablon)
- 12 SVG diagramů použitých v knize

Ukázky kódu z této knihy můžete volně používat ve svých projektech bez nutnosti atribuce. Pokud je zveřejníte nebo sdílíte, oceníme zmínku o zdroji, ale není to podmínkou.

Zpětná vazba a opravy

Tato kniha není dokonalá — žádná technická kniha v oblasti AI nemůže být, protože oblast se vyvíjí rychleji, než je možné tisknout. Pokud najdete technickou chybu, zastaralý příkaz nebo jiný problém, dejte nám vědět.

Uvítáme také zpětnou vazbu na to, co v knize chybí, co bylo nejužitečnější, nebo jaké konkrétní BC+AI problémy řešíte a chtěli byste v budoucím vydání pokrýt.

Kontakt: milos.mikulasek@ai4bc.dev

Poděkování

Tato kniha by ve svém rozsahu a tempu nevznikla bez nástrojů, které jsou zároveň jejím hlavním tématem.

Velký dík patří společnosti **Anthropic** za Claude — a konkrétně **Claude Code** a **Claude.ai**, které byly přítomny v každé fázi vzniku: od prvního návrhu struktury přes psaní a přepisování kapitol až po generování diagramů, ukázek kódu a sestavení samotného EPUB souboru.

Část textu, který právě čtete, prošla desítkami iterací v dialogu s AI — a výsledek je lepší, než by byl bez ní.

Je to trochu paradoxní situace: kniha o tom, jak AI mění práci vývojářů a konzultantů, byla sama napsána s pomocí AI. Snad právě to je nejpoctivější důkaz, že to, co popisujeme, opravdu funguje.

Obsah knihy

Modul 0: Úvod a orientace

1. Proč AI mění ERP svět (a proč to není hype)
2. AI nástroje pro BC vývojáře — praktický přehled
3. Realistická očekávání — co AI umí a co ne

Modul 1: AI jako vývojářský part'ák

1. Přehled AI asistentů — Claude, ChatGPT, Copilot, Gemini
2. Prompt engineering — základy pro vývojáře
3. Prompt engineering pro BC kontext — šablony
4. Claude Code — setup a konfigurace
5. Claude Code — práce s AL projekty
6. CLAUDE.md — kontext pro BC projekty
7. GitHub Copilot — setup ve VS Code pro AL
8. GitHub Copilot — efektivní použití s AL
9. Workshop — AL extension s AI, část 1: zadání → design
10. Workshop — AL extension s AI, část 2: implementace → review
11. AI pro testování AL kódu

Modul 2: AI pro denní práci konzultanta

1. AI pro funkční specifikace — od poznámek k dokumentu
2. AI pro testovací scénáře
3. AI pro analýzu zákaznických požadavků
4. AI pro dokumentaci — uživatelská i technická
5. AI pro komunikaci se zákazníkem
6. AI pro troubleshooting a debug BC problémů
7. AI + SQL pro analýzu BC dat
8. Shrnutí modulu — tvůj osobní AI toolkit
9. AI pro migraci dat do Business Central

Modul 3: AI integrace v Business Central

1. Microsoft Copilot v BC: co umí, jak nastavit, jak rozšířit
2. Azure OpenAI Service: přehled a setup
3. Volání AI API z AL kódu: HttpClient, JSON a REST
4. Demo: Automatická kategorizace položek pomocí AI
5. Demo: AI-powered vyhledávání v BC
6. Demo: Inteligentní návrhy pro prodejní objednávky
7. Power Platform a AI Builder
8. Copilot Studio a Business Central integrace
9. Bezpečnost, GDPR a etika při práci s AI
10. Jak prodat AI řešení zákazníkovi

Modul 4: Pokročilé techniky

1. RAG: Jak naučit AI tvoje BC data
2. AI agenti: Automatizace komplexních BC procesů
3. Fine-tuning vs. prompt engineering: kdy co použít
4. MCP servery a custom tooling pro BC
5. Emerging nástroje a trendy pro BC vývojáře
6. AI roadmapa pro BC vývojáře: kam to směřuje

Modul 5: Akční plán

1. 30denní plán implementace AI do tvé práce
2. Jak přesvědčit šéfa nebo firmu, aby investovala do AI
3. Jak se dál vzdělávat a nezůstat pozadu
4. Závěr: od vědění k dělání

Přílohy

A. Kompletní reference AL kódu B. Knihovna prompt šablon pro BC C. Cvičení ke kapitolám

O autorovi

Miloš Mikulášek pracuje s NAV a Business Central více než dvacet let — jako vývojář, školitel a člověk, který prošel každou fází ekosystému od prvních verzí Navision po dnešní cloudový BC. Vedle AL ovládá SQL na úrovni, která mu umožňuje řešit problémy tam, kde standardní BC nástroje nestačí.

Umělou inteligenci přijal jako přirozené rozšíření svého nástrojového arzenálu. Nezajímal ho hype — zajímalo ho, co AI skutečně dokáže v prostředí, které zná do hloubky. Výsledkem je tato kniha — *Business Central v době AI* — a komunita kolem ní.

Jeho přístup je přímý: žádné sliby, které AI nesplní, žádné zkratky tam, kde je potřeba skutečná znalost systému. Jen konkrétní nástroje a reálné příklady z praxe.

Najdeš ho na ai4bc.dev nebo na milos.mikulasek@ai4bc.dev.

Část I: Úvod a orientace

Proč AI mění ERP svět a jaké nástroje existují

Lekce 0.0: Proč AI mění ERP svět (a proč to není hype)

Před dvěma lety jsem na jedné BC konferenci stál u kávy s kolegou vývojářem. Byl zkušený — 12 let v BC, znalost NAV od verze 3.60. A říkal: „AI? To je jen hype. Za rok o tom nikdo nebude mluvit.”

Dnes ten člověk pracuje ve firmě, která mu nařídila „implementovat AI do BC do konce roku.” A on stojí před tabulí plnou post-itů a neví kde začít.

Nechci, aby se ti tohle stalo.

Tohle není lekce o tom, jak AI změní vše a jak musíš okamžitě změnit vše, co děláš. Tahle lekce ti dá kontext. Realistický, praktický kontext. Co se skutečně mění, proč to ovlivňuje práci BC vývojářů a konzultantů specificky, a proč je právě teď ten správný čas začít — ne proto, že to říká marketing, ale proto, že data ukazují konkrétní věci.

Pojďme si být upřímní ohledně toho, jak AI obsah na internetu většinou vypadá.

Na jedné straně jsou nadšenci: „AI nahradí vývojáře!” „AI změní vše do 5 let!” „Prompt engineering je nová superschopnost!”

Na druhé straně jsou skeptici: „Jenom hype.” „Halucinace jsou problém.” „Moji zákazníci AI nechtějí.”

A pak se vrátíš k práci, otevřeš VS Code, díváš se na AL kód, na codeunit, který musíš dopsat do pátku — a říkáš si: „Tak co teď? Jak mi tohle konkrétně pomůže?”

Žádný z těch článků ti neřekl, jak napsat codeunit s AI asistencí. Jak AI popsat BC architekturu. Jak pracovat s CLAUDE.md na reálném BC projektu. Jak integrovat Azure OpenAI do AL kódu tak, aby to bylo prakticky nasaditelné.

To je problém, který tato kniha řeší. A tahle lekce je začátek — dám ti framework, který ti pomůže pochopit, kde jsme, kam jdeme a co to znamená konkrétně pro tebe.

Co se skutečně mění — data, ne hype

Začneme čísly. GitHub provedl studii produktivity vývojářů s GitHub Copilotem — zkoumal 95 vývojářů na konkrétních úkolech. Zjistil průměrné zrychlení o 55 % u rutinních programátorských úkolů. 55 procent. To není zanedbatelné číslo.

McKinsey ve své studii z roku 2023 uvádí, že AI může zvýšit produktivitu vývojářů o 20 až 45 procent. A to mluvíme o úkolech jako boilerplate kód, dokumentace, testování, code review.

Ale pozor — tyhle studie jsou na obecném kódu. JavaScript, Python, Java. Jak je to s AL a Business Central?

Proč BC vývojáři mají výhodu i nevýhodu

Pojďme být konkrétní. Business Central má v kontextu AI specifické postavení.

Nevýhoda: AL není v tréninkových datech AI modelů zastoupen tak dobře jako hlavní programovací jazyky. Pokud se zeptáš AI „napiš mi AL codeunit pro X” bez kontextu, dostaneš výsledek, který bude vypadat syntakticky podobně, ale bude mu chybět znalost BC konvencí, event architektury, table relationships. AI nezná specifika BC jako DataPerCompany, jak fungují extensions vs. base app, jak správně subscribovat na eventy.

Výhoda: A tady přichází ta dobrá zpráva. Protože ty víš, co AL a BC znamenají, máš obrovskou výhodu oproti komukoli, kdo AI bez tohoto kontextu používá. Ty víš, co AI vygeneroval správně a co špatně. Ty rozumíš architektuře. Ty jsi expert v doméně — AI je tvůj nástroj.

● **Poznámka:** Dobrý BC vývojář s AI bude produktivnější než průměrný BC vývojář bez AI. A bude výrazně produktivnější než výborný obecný vývojář s AI, ale bez BC znalostí. Vaše doménová znalost je klíčová — AI ji zesiluje, nenahrazuje.

Tři vlny AI v ERP světě

Rád přemýšlím o tom, kde jsme, pomocí metafory tří vln. Pomůže ti pochopit, kde jsme teď a co přichází.

Vlna 1 — AI asistenti pro vývojáře. To jsou nástroje jako ChatGPT, Claude, GitHub Copilot. Pomáhají psát, vysvětlovat a ladit kód, generovat dokumentaci, odpovídat na technické dotazy. Tady jsme teď. Tahle vlna je reálná, dostupná, použitelná dnes. A přesně to pokrývá tato kniha v modulech 1 a 2.

Vlna 2 — AI přímo zabudovaná do produktu. Microsoft Copilot v Business Central, AI-powered vyhledávání, automatické návrhy pro uživatele, inteligentní workflows. Tady začínáme být. Microsoft to aktivně rozvíjí a v posledních verzích BC vidíme první funkce. Modul 3 knihy se na tohle zaměřuje.

Vlna 3 — Autonomní AI agenti. Systémy, které samy monitorují procesy, rozpoznávají anomálie, navrhnou a provádějí akce v BC. Tahle vlna přijde v horizontu 2 až 3 let. V modulu 4 se o ní bavíme, protože chceme, abyste měli kontext — i když ji ještě plně implementovat nedokážeme.

Proč ti to říkám? Protože pochopit, kde na timeline jsme, ti pomůže nastavit správná očekávání. Vlna 1 a 2 jsou tady a fungují. Vlna 3 přijde — a ti, kdo se připravují dnes, budou připraveni zítra.

Co to znamená pro tvoji kariéru

Řeknu ti, co si myslím upřímně. Dobrá zpráva: Znalost BC a AL zůstává klíčová. AI ji v dohledné době nenahradí. Zákazník pořád potřebuje někoho, kdo rozumí jeho businessu, kdo mu navrhne řešení, kdo nese zodpovědnost za výsledek.

● **Upozornění:** Vývojář, který AI nepoužívá, bude dělat za celý den to, co vývojář s AI udělá za dopoledne. To neznamená, že vás AI nahradí — ale může vás nahradit vývojář, který AI používá lépe.

To neznamená, že tě AI nahradí. Znamená to, že tě může nahradit vývojář, který AI používá lépe než ty. A ten vývojář nemusí být tak zkušený, jak jsi ty — protože AI mu pomáhá překlenout část rozdílu v zkušenostech.

Takže: Nejde o to, jestli AI používat. Jde o to, jak ho používat dobře. A to je přesně to, co se v této knize naučíš.

Shrnutí

Pojďme si shrnout, co jsme si dnes řekli.

Za prvé: AI v BC vývoji není hype. Jsou za tím reálná čísla — 55 % produktivita u GitHub Copilot studie, 20–45 % zvýšení produktivity dle McKinsey.

Za druhé: BC vývojáři mají specifickou výhodu. Rozumíme kontextu, který AI sám nezná. Jsme experti v doméně — AI je nástroj, který tuhle expertízu zesiluje.

Za třetí: Jsme ve vlně 1 — AI asistenti jsou zde dnes, dostupní a použitelní. Vlna 2 (AI v BC produktu) začíná. Vlna 3 (autonomní agenti) přichází.

Za čtvrté a nejdůležitější: Nejde o otázku „jestli“. Jde o otázku „jak dobře“. A to je přesně to, na co se zaměřuje zbytek této knihy.

Konkrétní příklady: AI v BC praxi dnes

Abychom nebyli jen u čísel a studií, pojďme si ukázat konkrétní situace z praxe BC specialistů, kteří AI již používají.

Příklad 1: Vývojář a boilerplate kód

Vývojář dostane zadání: „Potřebujeme tabulku pro evidenci servisních smluv s poli pro zákazníka, datum začátku, datum konce, měsíční poplatek, stav a poznámku. K tomu stránku pro správu a codeunit pro automatickou kontrolu expirace.“

Bez AI: Otevře VS Code, založí tabulku, pole po poli. Pak stránku, pole po poli. Pak codeunit. Celkem 45–60 minut rutinní práce, kde se nedělá žádné rozhodnutí — jen se píše standardní struktura.

S AI: Zadá prompt s požadavky, konvencemi (prefix, komentáře) a za 5 minut má kompletní tabulku, stránku i základ codeunitu. Dalších 15 minut stráví review a přizpůsobením. Celkem 20 minut. Úspora: 30–40 minut na jednom úkolu.

Příklad 2: Konzultantka a zákaznický workshop

Konzultantka má zítra discovery workshop u zákazníka. Má téma (úprava expedičního procesu), ale žádnou agendu. Dříve by strávila hodinu přípravou.

S AI za 15 minut: strukturovaná agenda workshopu, klíčové otázky seřazené podle priority, seznam dokumentů, které zákazník má přinést, a přehled typických problémů BC v této oblasti. Přijde na workshop připravená a zákazník to vidí.

Příklad 3: Freelancer a nabídka

Freelancer dostal poptávku na BC extension pro správu nájmu. Potřebuje napsat nabídku s odhadem, rozsahem a cenou.

S AI: zadá popis požadavků a AI rozloží projekt na části, odhadne hodiny pro každou část, identifikuje rizika a navrhne fáze implementace. Freelancer nabídku upraví, doplní svou sazbu a za 30 minut má profesionální dokument — místo 2 hodin přemýšlení a formátování.

Co tyto příklady mají společného

Ve všech třech případech AI nedělá rozhodnutí — člověk rozhoduje. AI dělá rutinní práci: strukturování textu, generování boilerplate kódu, formátování dokumentu. Člověk přidává expertní znalost: BC architektura, zákaznické specifiky, cenová strategie.

Tohle je model, který funguje. A přesně tenhle model vás naučí zbytek knihy.

Vyzkoušejte si

Cvičení 1: Měření aktuálního stavu. Tento týden si poznamenejte 3 úkoly, které děláte, a kolik minut na nich strávíte. Na konci týdne budete mít benchmark pro porovnání s AI.

Cvičení 2: První kontakt s AI. Pokud jste AI ještě nezkusili: otevřte claude.ai nebo chatgpt.com a zadejte: „Jsem BC vývojář. Napiš mi jednoduchou AL proceduru pro kontrolu, zda zákazník má vyplněný email.“ Funguje to? Co byste změnili?

Přehled AI nástrojů pro BC vývojáře

Nedávno mi kolega řekl, že strávil celý víkend testováním AI nástrojů. Claude, Cursor, Codeium, Perplexity, ChatGPT... Nainstaloval je, vyzkoušel, přečetl si recenze.

V pondělí přišel do práce a nevěděl, který z nich vlastně používat pro svůj BC projekt. Vrátil se k Googlu.

Tohle je přesný opak toho, co chceme. Cílem není mít co nejvíce nástrojů. Cílem je mít ten správný nástroj pro správnou situaci a naučit se ho používat dobře.

Problém není nedostatek AI nástrojů. Je jich příliš mnoho — a každý týden přibývají nové. ChatGPT, Claude, Gemini, Copilot, Cursor, Codeium, Tabnine, Perplexity... každý s vlastní marketingovou kampaní o tom, jak je nejlepší.

Pro BC vývojáře to vytváří paralýzu analýzy. Kolik nástrojů potřebuji? Musím platit za všechny? Co používá zbytek BC komunity?

Odpověď je jednodušší, než si myslíš. Existují tři kategorie nástrojů, které jsou pro BC práci relevantní. V každé kategorii je jeden nebo dva jasní vítězové. A na konci kapitoly dostaneš decision framework — jedno jednoduché pravidlo, podle kterého poznáš, co pro konkrétní situaci použít.

Kategorie 1: General-purpose AI asistenti

Tato kategorie zahrnuje AI nástroje, se kterými komunikuješ přes chat rozhraní. Píšeš dotazy, dostáváš odpovědi, zadáváš kód ke review, prosíš o napsání dokumentace. Jsou to víceúčelové nástroje.

Claude od Anthropic je momentálně můj hlavní nástroj pro BC práci. Proč? Má jeden z největších context window na trhu — to znamená, že mu dokážeš dát celý obsáhlý AL soubor nebo celou specifikaci a on s tím pracuje konzistentně. Ale hlavní výhodou je kvalita výstupu pro AL kód a iterativní práce s projektem přes Claude Code. Pro psaní funkčních specifikací, analýzu požadavků, dokumentaci a generování AL kódu s dobrým promptem je výborný. Zdarma dostaneš základní verzi, Claude Pro za 20 dolarů měsíčně ti dá přístup k lepšímu modelu a vyšším limitům. A Claude Code — o tom si povíme v Modulu 1 — je celý vývojový asistent postavený na Claude.

ChatGPT od OpenAI byl první masově rozšířený AI asistent a stále je velmi dobrý. Pro rychlé dotazy, vysvětlení konceptů a obecné kódování funguje skvěle. Pro BC specifickou práci je Claude o něco lepší kvůli většímu context window, ale ChatGPT je platná alternativa — zvláště pokud ho už platíš nebo ho máš přes firemní licence.

Gemini od Googlu je zajímavý hlavně pro ty, kdo pracují v Google Workspace ekosystému. Pro kódování je v současnosti o krůček za Claude a ChatGPT, ale pro analýzu dokumentů a vyhledávání v reálném čase má zajímavé funkce.

Verdikt pro BC práci: Doporučuji Claude jako primární nástroj. ChatGPT jako záloha nebo pokud ho máš firemně.

Kategorie 2: Vývojářské AI asistenti (přímo v editoru)

Tohle je kategorie, kde se AI integruje přímo do VS Code nebo jiného editoru a pomáhá ti při psaní kódu v reálném čase.

GitHub Copilot je nejrozšířenější nástroj v téhle kategorii. Pracuje přímo ve VS Code, vidí aktuálně otevřené soubory a navrhuje doplnění kódu. Pro obecné programování je výborný. Pro AL výrazně záleží na kontextu — pokud má Copilot viditelné dobré AL soubory ve workspace, návrhy jsou lepší. Od listopadu 2024 existuje **Free tier** (2 000 dokončení + 50 chat zpráv měsíčně), který stačí pro první experimenty s BC. Placený Individual plán stojí 10 dolarů měsíčně; pro studenty a open source projekty je zdarma. V Modulu 1 se na setup a efektivní používání pro AL podíváme detailně.

Claude Code je trochu jiný typ nástroje — není to jen inline doplňování kódu. Je to fullstack AI vývojový asistent. Čte celý projekt, provádí změny v souborech, spouští příkazy, rozumí git historii. Říkám tomu „AI vývojář, který sedí vedle tebe a vidí vše, co vidíš ty.“ Pro AL projekty je to momentálně nejsilnější volba. Zhrnuto v Claude Pro. Moduly 1.4 až 1.6 jsou celé věnovány Claude Code pro BC projekty.

Verdikt: Claude Code jako hlavní pro hlubší projektovou práci, GitHub Copilot jako doplněk pro inline návrhy při psaní.

Kategorie 3: Microsoft AI ekosystém pro BC

Tady se dostáváme na území, které je specificky pro BC vývojáře a konzultanty.

Microsoft Copilot v Business Central je AI asistent přímo zabudovaný v BC uživatelském rozhraní. Pomáhá uživatelům vyhledávat záznamy, vysvětluje pole a funkce, navrhuje akce. Z pohledu vývojáře je zajímavé, že tento Copilot lze rozšiřovat vlastními rozšířeními — a to je velká příležitost pro BC partnery. Zákazníci to budou chtít. Podrobně v Modulu 3.

Azure OpenAI Service je to, co budeš potřebovat, když chceš AI integrovat přímo do BC řešení pro zákazníka. Je to GPT model přístupný přes API, hostovaný na Azure. Výhody oproti přímému volání OpenAI API: je to enterprise-grade, GDPR compliant, máš SLA, data ti nezůstávají uložena u poskytovatele. Pro produkční BC řešení je toto standardní volba. V Modulu 3 ti ukážu, jak ho volat přímo z AL kódu přes HttpClient, s funkčními příklady.

AI Builder v Power Platform je no-code cesta k AI funkcím. Rozpoznávání dokumentů, sentiment analýza, predikce, extrakce dat z faktur. Bez kódu, přes Power Apps nebo Power Automate. Pro BC konzultanty, kteří nechtějí jít cestou AL kódu, je to reálná alternativa. Taky Modul 3.

Decision framework — kdy použít co

Pojďme si to shrnout do jednoduchého pravidla. Tady je tabulka, která pomůže v situacích každodenní práce.

Situace	Doporučený nástroj
Píšu AL kód a chci AI jako parťáka pro celý projekt	Claude Code
Chci inline doplňování kódu přímo při psaní v VS Code	GitHub Copilot
Rychlý dotaz, vysvětlení, debug jednoduchého problému	Claude nebo ChatGPT
Píšu funkční specifikaci nebo dokumentaci	Claude
Integruji AI do BC řešení pro zákazníka (produkce)	Azure OpenAI Service
Zákazník chce AI funkce bez vývoje v AL	AI Builder / Copilot Studio
Chci rozšířit Microsoft Copilot v BC	Copilot Extensions (AL)

Tenhle framework platí na 80 % situací. Výjimky budou, ale pro začátek tohle postačí.

Shrnutí

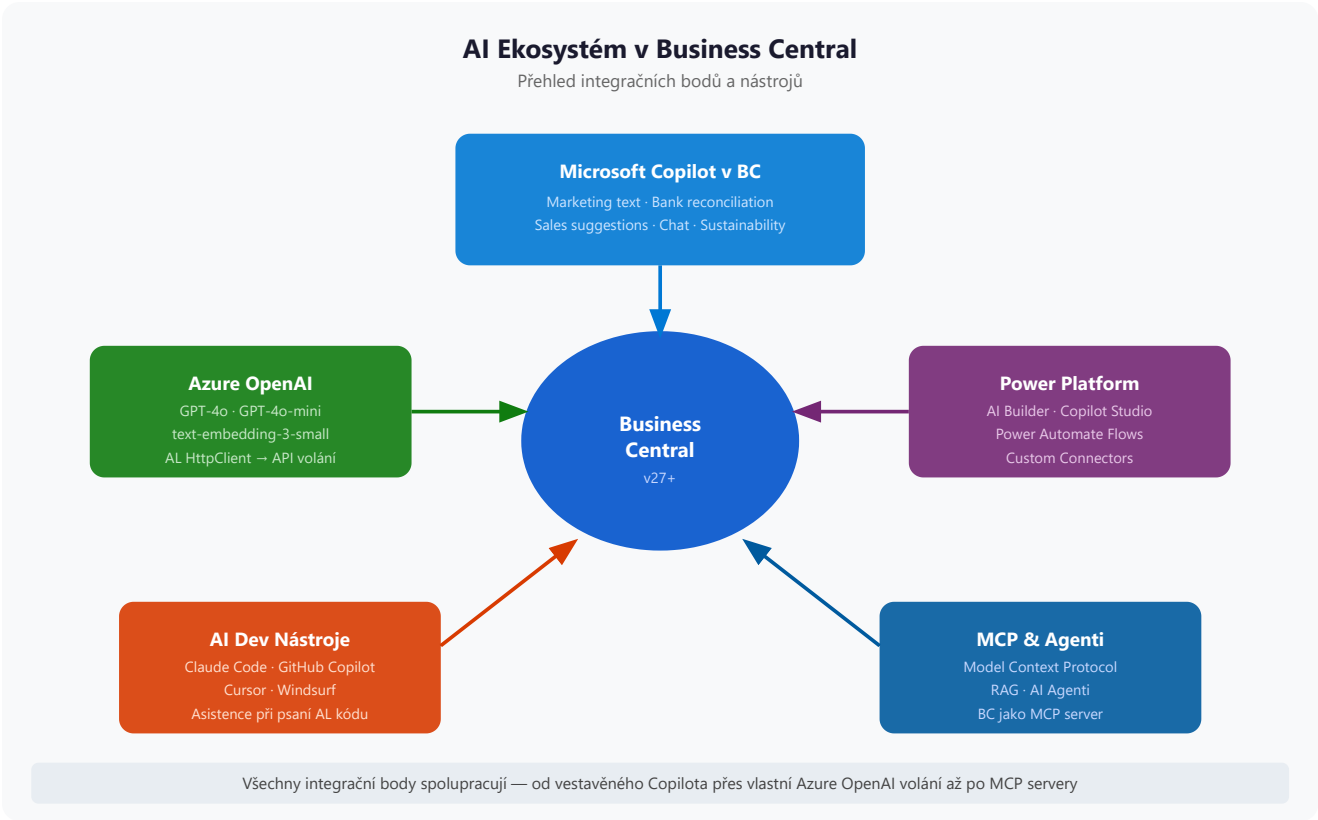
Tři kategorie AI nástrojů pro BC práci.

Kategorie 1 — General-purpose asistenti: Claude jako primární, ChatGPT jako alternativa. Oba přes chat rozhraní, výborné pro dokumentaci, specifikace a kód s dobrým promptem.

Kategorie 2 — Vývojářské asistenti v editoru: Claude Code pro komplexní projektovou práci, GitHub Copilot pro inline návrhy. Tyhle dva nástroje ti budou nejbližší při každodenním vývoji.

Kategorie 3 — Microsoft ekosystém: Copilot v BC a Azure OpenAI pro produkční integrace do BC řešení. Power Platform pro no-code přístupy.

Doporučený starter stack: Claude Pro za 20 dolarů měsíčně a GitHub Copilot za 10 dolarů měsíčně (nebo začni s Free tierem). Za 30 dolarů měsíčně — méně než polodenní fakturace — máš plnohodnotný AI toolkit pro BC vývoj. Pro intenzivní uživatele Claude Code existuje **Claude Max plán** (100 dolarů měsíčně) s výrazně vyššími limity — pro profesionály, kteří AI používají na většinu své kódové práce, je obvykle lepší volba.



Obr. 1: Ekosystém AI nástrojů pro Business Central

Kdy použít který AI nástroj v BC projektu			
Rozhodovací matice pro BC vývojáře a konzultanty			
Nástroj	Nejllepší pro	Konkrétní BC use cases	Nevhodné pro
Claude Code / GitHub Copilot AI IDE asistent	Psaní a refaktoring kódu Debugging a vysvětlování Generování testů Code review	AL codeunits a page extensions Oprava compile chyb v AL Dokumentace procedur CLAUDE.md pro BC projekty	Produkční AI funkce v BC pro zákazníky
Azure OpenAI + AL HttpClient Custom integrace	Vlastní AI logika v BC Kategorizace a klasifikace Generování textu z BC dat Structured JSON output	Auto-kategorizace BC položek Návrhy prodejních položek AI-powered vyhledávání Inteligentní upozornění	Psaní kódu, generování dokumentace pro vývojáře
Microsoft Copilot v BC Vestavěný Copilot	Hotové AI funkce bez kódu Rychlé nasazení u zákazníka Standard Microsoft UX Minimální nastavení	Marketing text pro položky Bank reconciliation Návrhy řádků objednávek Sustainability tracking (BC 27)	Custom byznys logika, specifické firemní procesy
Power Platform AI Builder Low-code AI	Zpracování dokumentů Automatizace s AI bez kódu Chatboty pro zákazníky Prediktivní modely	Skenování faktur do BC Zákaznický bot pro stav obj. AI klasifikace z Power Automate Predikce poptávky	Komplexní custom kód, vysoký objem API volání
RAG / Agenti / MCP Server Pokročilé techniky	Komplexní znalostní systémy Autonomní procesy v BC Přímý přístup AI k BC datům	Chatbot nad BC dokumentací Agent pro zpracování emailů BC jako MCP server pro Claude	Jednoduché úkoly – zbytečně složitě řešení

Obr. 2: Rozhodovací matice – kdy použít který AI nástroj

Realistická očekávání — co AI umí a co ne

Jeden z mých kolegů — zkušený BC vývojář, 8 let praxe — strávil tři hodiny snahou přimět ChatGPT, aby mu napsal kompletní BC modul pro správu smluv.

Výsledek? Kód, který vypadal správně a zkompileval se. Ale zhruba polovina věcí nefungovala, jak měla. Relationships nebyly správně. Event subscribers chyběly. Naming konvence nevhodné pro jejich projekt. Strávil další hodinu opravováním a pak to vzdal.

Skončil frustrovaný a přesvědčený, že „AI na BC vývoj nestačí.“

Neměl pravdu. Ale jeho očekávání byla špatná. A to je přesně problém, který tato kapitola řeší — dřív než se dostaneme do praxe v Modulu 1.

AI nástroje jsou extrémně výkonné — ale jen pokud víš, na co je použít.

Když od nich čekáš víc než mohou dát, skončíš frustrovaný jako můj kolega. Budeš říkat „AI nefunguje“ — přestože problém byl ve způsobu použití, ne v nástroji.

Naopak, pokud od nich čekáš méně, necháš na stole obrovské množství ušetřeného času. Budeš AI používat jen na triviální úkoly, zatímco ho můžeš použít na věci, které ti ušetří hodiny.

Správná kalibrace očekávání je doslova rozdíl mezi „AI mi nic nedává“ a „AI mi ušetří 10 hodin týdně.“ Pojdme tu kalibraci udělat správně.

Co AI umí dobře — pro BC vývojáře a konzultanty

Boilerplate a rutinní kód. Codeunity pro validate, jednoduché výpočty, standardní CRUD operace, event subscribers pro standardní eventy, AL interfaces a jejich implementace — to AI zvládá výborně, pokud dostane dobrý prompt s kontextem. V Modulu 1 se naučíš přesně, jak takový prompt napsat.

Vysvětlování existujícího kódu. „Vysvětli mi, co dělá tenhle codeunit“ nebo „Proč by tenhle kód mohl způsobovat chybu X“ — to je oblast, kde AI exceluje. Hodí se při onboardingu na nový projekt, při code review, nebo prostě když dědíš kód po kolegovi, který ve firmě už není.

Dokumentování kódu. Přidat komentáře k existujícímu kódu, vygenerovat XMLdoc dokumentaci k procedurám, napsat README pro extension — to AI dělá rychle a dobře. Pro konzultanty: hodinová práce na dokumentaci se zkrátí na 10–15 minut.

Psaní funkčních specifikací a testovacích scénářů. Tohle je oblast, kde konzultanti ušetří nejvíc času. Zákazník ti řekne, co chce — ty to přepíšeš do poznámek — AI z toho vygeneruje strukturovanou funkční specifikaci. Nebo naopak: ze specifikace vygeneruje testovací scénáře. Tenhle workflow si projdeme detailně v Modulu 2.

Refactoring a code review. „Jak bych tenhle kód napsal lépe?“ nebo „Jaké jsou potenciální problémy v tomto kódu?“ To jsou dotazy, na které AI odpovídá kvalitně. Může upozornit na výkonnostní problémy, chybějící error handling, nebo nekonzistentní naming.

Debugging s popisem problému. „Dostávám tuto chybu v AL: [přesná chyba]. Tady je kód: [kód]. Co je špatně?“ Tohle funguje překvapivě dobře. Zejména u compile-time chyb a logických problémů, které jsou popsateľné textem.

Co AI neudělá dobře — nebo vůbec

Kompletní BC modul od nuly, bez detailního kontextu. Přesně to, co zkoušel kolega z úvodu. „Napiš mi modul pro správu smluv“ je příliš vágní. AI neví, jak tvoje firma smlouvy eviduje, jaká jsou business pravidla, jak se to integruje s ostatními moduly. Bez tohoto kontextu vygeneruje generické řešení, které vypadá správně, ale nesedí na tvůj projekt.

Řešení: Rozděluj. Každý codeunit zvlášť. S kontextem tabulek, business pravidel, naming konvence projektu.

Rozumět zákaznickově businessové logice. AI neví, že zákazník má 15 let starý způsob fakturace, který nikde není zdokumentovaný, ale všichni ho znají. AI neví o výjimkách, které „jsou takové od vždy“. AI neví o politických rozhodnutích („tenhle proces je takový, protože to chce ředitel“). To víš jen ty — a proto jsi jako BC odborník nenahraditelný.

Testovat kód v Business Central. AI napíše kód, ale nepustí ho. Nezkontroluje, jestli extension jde zkompileovat. Nejistí runtime chybu. Debugging v BC — to stále musíš ty. AI ti může pomoci navrhnout řešení, ale ověření reálného chování v BC prostředí je na tobě.

Nahradit znalost BC ekosystému. Toto je klíčové. Vývojář bez znalosti AL a BC dostane od AI stejně nepoužitelný výsledek jako bez AI. AI zesiluje tvoje znalosti, nenahrazuje je. Proto kniha, kterou teď čteš, předpokládá, že AL a BC základy znáš.

Garantovat aktuálnost informací. AI modely mají datum cutoff — datum, do kdy byl natrénován. Nejnovější BC verze, nové API funkce, aktuální best practices pro AL — vždy ověřuj v oficiální Microsoft dokumentaci. AI ti může říct, jak to fungovalo do verze 26, ale co přišlo ve verzi 27, nemusí vědět.

Zlaté pravidlo práce s AI v BC

AI je junior vývojář, který nikdy nepracoval v Business Central.

Je chytrý, pracovitý a rychlý. Přečte si vše, co mu dáš. Udělá, co mu řekneš. Ale nikdy neviděl reálný BC projekt. Nezná zákaznickovy procesy. Nemá intuici, která přichází s lety zkušeností v BC.

To znamená, že potřebuje: - Kontext projektu (proto CLAUDE.md — o tom v Lekci 1.5) - Jasné zadání s dostatkem detailů - Tvoji kontrolu výstupu - Korekci, když udělá chybu — a udělá ji

● **Upozornění:** Vždy kontroluj kód od AI, jako bys ho dostal od juniorního kolegy. Nikdy se nespolehej na to, že „to AI určitě udělal správně.“ Compile neznamena correct.

Nejčastější chyby, které vidím u začátečníků

Chyba 1: Příliš vágní prompt. „Napiš mi BC extension.“ AI neví co, proč, pro jaké tabulky, s jakou logikou. Výsledek: generický kód, který neodpovídá tvému projektu. Řešení: V Modulu 1, Lekce 1.2 a 1.3 se

naučíš prompt engineering specificky pro BC kontext.

Chyba 2: Nekontrolovat výstup. AI může vygenerovat kód, který zkompileje, ale dělá špatnou věc. Přepisuje záznamy, které přepisovat nemá. Volá tabulky, které neexistují v tvé BC verzi. Reviewuj kód jako od juniora.

● **Tip:** Příliš dlouhé konverzace snižují kvalitu výstupu — AI „zapomíná“ kontext z začátku. Pro nový úkol vždy začni novou konverzací s čerstvým kontextem.

Shrnutí

AI v BC práci funguje výborně pro boilerplate kód, vysvětlování a dokumentaci, specifikace a testovací scénáře, refactoring, debugging s popisem. To jsou oblasti, kde ušetříš hodiny každý týden.

Nečekej od AI kompletní řešení bez kontextu, znalost zákaznickovy businessové logiky, testování v BC nebo záruky aktuálnosti.

Zlaté pravidlo: Junior vývojář bez BC zkušeností. Dávej mu kontext, kontroluj výstup, koriguj chyby.

Praktický test: Kde AI ušetří čas a kde ne

Provedli jsme neformální průzkum mezi 15 BC vývojáři a konzultanty, kteří AI aktivně používají. Otázka: „Kde vám AI ušetří nejvíc času a kde je k ničemu?“

Top 5 oblastí úspory času (seřazeno podle hlasů)

1. **Generování boilerplate AL kódu** (14/15 hlasů) — tabulky, stránky, event subscribers, gettery/settery. AI zvládá perfektně, protože je to strukturovaný vzor.
2. **Psaní dokumentace a specifikací** (13/15) — struktura, formulace, konzistence. AI drafty jsou z 80 % použitelné.
3. **Vysvětlování neznámého kódu** (12/15) — starý kód, cizí extension, nevysvětlený modul. AI čte a vysvětluje rychleji než Stack Overflow.
4. **SQL dotazy** (11/15) — BC tabulková struktura je obsáhlá. AI zná správné názvy tabulek a polí, generuje funkční dotazy.
5. **Debugging s chybovými hláškami** (10/15) — přesný text chyby + kontext = rychlá diagnóza. Lepší než Google.

Top 5 oblastí, kde AI nepomáhá (seřazeno)

1. **Zákaznická business logika** (15/15 hlasů) — AI nezná zákazníka, jeho procesy, jeho výjimky. Tohle víte jen vy.
2. **Testování v BC prostředí** (14/15) — AI napíše kód, ale nepustí ho v BC. Runtime verifikace je na vás.

3. **Architektonická rozhodnutí v kontextu zákazníka** (12/15) — „Máme to udělat jako extension nebo modifikaci?“ závisí na zákaznické situaci.
4. **Řešení licenčních a obchodních otázek** (11/15) — AI neví, jaký pricing váš partner nabízí.
5. **Debugging race conditions a intermitentních chyb** (10/15) — problémy, které se nedají popsat textem, potřebují hands-on diagnostiku.

Všimněte si vzoru: AI je silná na strukturované, opakovatelné úkoly. Slabá na kontextové rozhodování, které vyžaduje znalost zákazníka nebo prostředí.

Vyzkoušejte si

Cvičení 1: Vlastní seznam. Napište 5 úkolů z vaší práce, které děláte každý týden. U každého odhadněte: ušetří AI 50 %+ času? 10–50 %? Nebo nepomůže?

Cvičení 2: Test zlatého pravidla. Zadejte AI úkol z vaší práce jako byste ho zadávali juniornímu kolegovi. Jaké instrukce musíte přidat, aby výsledek byl použitelný? To jsou přesně instrukce, které patří do promptu.

Vážení čtenáři, právě jste dočetli ukázkou z knihy Business Central v době AI.
Pokud se Vám ukázka líbila, na našem webu si můžete zakoupit celou knihu.