

Tvorba a integrácia IT riešení — zaujímavosti z praxe

Ing. Vladimír Mlynarovič, PhD.
Prednáška na FIIT STU v predmete PSI

7. máj 2026





Agenda

Agenda

1

O mne a tejto prednáške

2

DXC & Enterprise svet

3

Od Zadania (RFP) k Zákazke

4

Manažment Ponuky -
Zákazky

Diskusia a Prestávka

Agenda

5

Tvorba a integrácia riešenia

6

Životný cyklus riešenia

7

Cenotvorba

8

Optimalizácia riešenia

Diskusia a Prestávka

Agenda

9

Manažment predpokladov,
závislostí a rizík

10

Od Ponuky cez Zmluvu ku
Dielu

Diskusia a Záver

Agenda

1

O mne a tejto prednáške

2

DXC & Enterprise svet

3

Od Zadania (RFP) k Zákazke

4

Manažment Ponuky -
Zákazky

Diskusia a Prestávka



1

O mne a tejto prednáške

Úvod a kontext

- Kto som a čo robím v DXC Technology
 - čo má tvorba a integrácia IT riešení spoločné s predmetom PSI
 - architektúra, integrácia, náklady, predpoklady, riziká, ponuka, zmluva
 - prečo UML, procesy a dokumentácia existujú
 - od RFP po prevádzku
-
- 150 min ≠ monológ
 - otázky počas prednášky vítané
 - prednáška + diskusia + praktické ukážky.

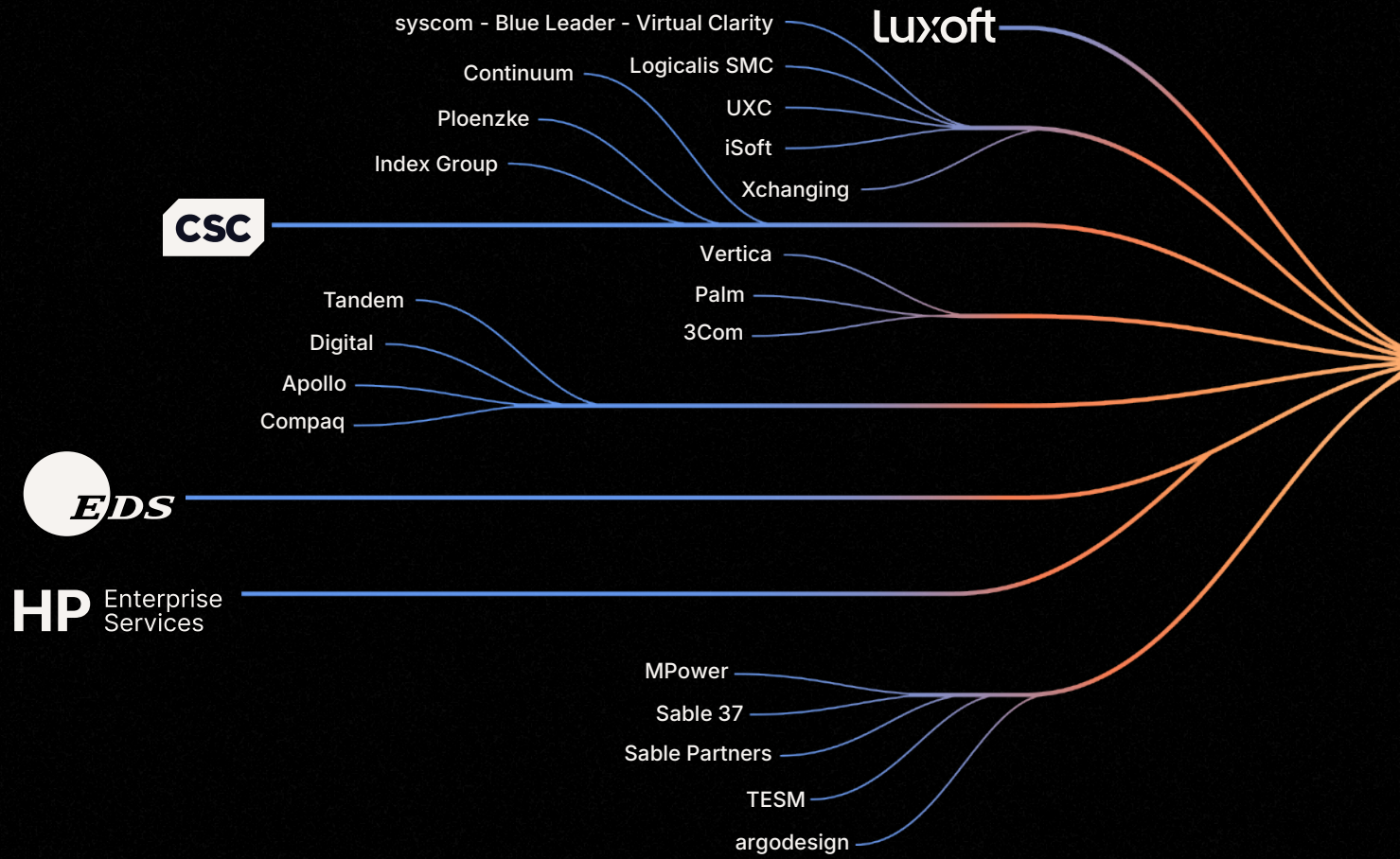


2

DXC & Enterprise svet

Trusted heritage

SIX DECADES OF MISSION-CRITICAL INNOVATION...



Industry Platforms

Sector-transforming software that modernizes critical operations at scale



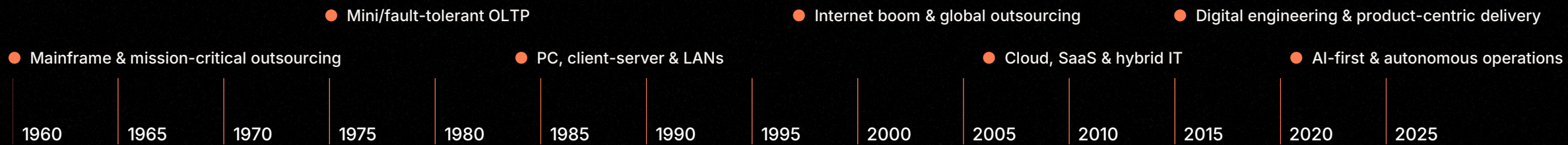
Services

Full life-cycle technology services that build, run, and protect enterprise systems



Solutions

Integrated outcomes that turn complexity into clarity and speed into results



DXC AT A GLANCE

60+

years of innovating



\$12.9B

FY25 revenue



#315

2025 Fortune 500



100

2025 Disability Index



115K

employees



We run today's critical systems and build tomorrow's breakthroughs

70K

workload migrations per year



230+

partners



70+

countries across the globe



230+

customers in Fortune 500



40K

developers



Integrované portfólio a proces

Build & Innovate

Advisory	We align strategy, technology, and operations to accelerate measurable outcomes.
Application Dev & Modernization	We design, build, and streamline tailored software that fits unique business workflows.
Customer Experience & Product Design	We craft user-centered products that deepen engagement, build loyalty, and deliver exceptional experiences.
Data & AI	We unlock real-time insights with production-ready AI.
Enterprise Applications	We modernize and manage core ERP and CRM platforms to drive efficiency and scale operations.

Manage & Optimize

Cloud & Infrastructure	We migrate, optimize, and run hybrid clouds for peak resilience and cost control.
Managed Applications	We integrate tailored software that fits unique business needs then manage and optimize it for continuous performance.
Business Process Services	We streamline policy, claims, and operations with proven platforms that reduce complexity and costs.
Modern Workplace	We enable seamless, secure collaboration and productivity from any device, anywhere.

Protect & Scale

Cybersecurity	We protect critical assets end to end with adaptive, zero-trust defenses that evolve with threats.
Infrastructure Protection	We secure and monitor critical infrastructure with adaptive, intelligent defenses that respond to emerging threats.
Modernization as a Service (MaaS)	We transform legacy systems into agile, cloud-ready platforms through continuous modernization and strategic optimization.



Discover

Define the blueprint for success

Typical Outcome
Strategic roadmap in weeks with prioritized quick wins



Develop

Engineer solutions tailored to your business needs

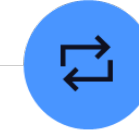
Typical Outcome
MVP deployment in 8-12 weeks using AI accelerators



Deploy

Launch with confidence using proven models

Typical Outcome
40% faster go-live with pre-built integration frameworks



Operate

Deliver continuous value with automation and optimization

Typical Outcome
30-50% operational cost reduction through intelligent automation



3

Od Zadania (RFP) k Zákazke

RFP: čo to je a prečo je zradné

- RFI (Request for Information) = vyžiadanie portfólia, capabilít, certifikátov, životopisov, referencií.
- RFP (Request for Proposal) = zadanie + pravidlá hry (scope, kritériá, termíny, formát odpovede)
- „Shall/Must“ = povinné → nesplnenie môže byť diskvalifikácia („knock-out“)
- Ticho v RFP ≠ voľnosť → vznikajú predpoklady
- Riziko nejasností: scope nejasný ⇒ vysoká volatilita & cena

Typy požiadaviek: funkčné vs nefunkčné

- **Funkčné** (správne skôr **Funkcionálne**) požiadavky (**FR**): *čo systém robí* (funkcie, procesy, výstupy)
- **Nefunkčné** (správne skôr **Nefunkcionálne**) požiadavky (**NFR**): *ako to musí fungovať* (SLA/KPI, bezpečnosť, regulácie, prevádzka)
- Príklady NFR z praxe:
 - **SLA / Business KPI commitments** – musia byť v akceptovateľných limitoch
 - **Regulačné obmedzenia pre delivery model**

Analýza RFP: ako to robíme prakticky

1. **Vyextrahovať všetky požiadavky** – vhodné použiť nástroj, checklist
2. **Určiť rozsah & mieru pokrytia**
3. **Zoznam predpokladov a rizík** (napr. „Top 10“)
4. **Overiť vstupy na odhad**: baseline/sizing dáta od klienta (častokrát chýbajú)
5. **Skontrolovať NFR a Zmluvu**: SLA/KPI limity, T&Cs mapovanie na Zmluvu, komerčný model, platobné míľniky

Q&A fáza: dobré otázky = lacnejší projekt

- Cieľ Q&A: znižovať neistotu ⇒ lacnejší odhad / menšia rezerva / menej prepracovaní
- Otázky typicky šetria peniaze, keď sa týkajú:
 - chýbajúcich baseline dát (sizing, objemy, inventár)
 - predpokladov a rizík (aby neboli „na guess level“)
 - due diligence / joint verification fázy (navrhnuť, ak treba)
 - SLA/KPI a penalty mechanizmov (napr. „earn back mechanism“)

RFP súlad: „shall/must“ mindset

- **Zmapovať všetky “Shall/Must”** → Plne adresovať + prípadný odkaz na detailnú odpoveď
- Identifikovať „**knock-out (KO) kritériá/otázky**“ a odpovedať striktne v súlade
- Keď RFP niečo nedefinuje → vytvoriť **explicitné predpoklady**
- **SLA mindset**: držať záväzky v rozumných limitoch + ošetriť systém pokút

Mini-ukážka pre analýzu RFP/požiadavky

„Riešenie musí zabezpečiť spracovanie požiadaviek používateľov naprieč viacerými systémami, byť vysoko dostupné a spĺňať bezpečnostné štandardy Zákazníka.“

Vytvorte 3 otázky, ktoré optimalizujú náklady a znižujú riziko

Mini-ukážka pre analýzu požiadavky

„Riešenie musí zabezpečiť spracovanie požiadaviek používateľov naprieč viacerými systémami, byť vysoko dostupné a spĺňať bezpečnostné štandardy Zákazníka.“

„V PSI by sme povedali: toto nie je jedna požiadavka. Je to zmes viacerých konceptov, ktoré treba oddeliť.“

- **spracovanie požiadaviek => procesný model**
- **viaceré systémy => integrácia**
- **vysoká dostupnosť => NFR / SLA**
- **bezpečnostné štandardy => NFR / Bezpečnosť**

„Presne toto robíte v PSI – rozkladáte realitu na model.“

1

Procesná otázka

„Aký presne proces spracovania požiadaviek – kroky, výnimky, objemy?“

=> bez toho nevieme rozsah ani prácnosť

2

Integračná otázka

„Ktoré systémy sú zapojené, aké rozhrania existujú a kto ich vlastní?“

=> bez toho nevieme architektúru ani zodpovednosti

3

NFR otázka

„Čo presne znamená vysoká dostupnosť a ktoré security štandardy sú povinné?“ / merateľnosť

=> tu sa najčastejšie láme cena aj riziko

Go / No-Go rozhodnutie

Na základe analýzy RFP vieme urobiť **Go / No-Go rozhodnutie**

Typické **NO-GO** signály:

- scope je vágny alebo si protirečí
- RFP má tvrdé požiadavky, ktoré vieme splniť len s extrémnym rizikom
- SLA alebo komerčný model by zabil maržu
- nemáme čím sa odlíšiť → nízka pravdepodobnosť úspechu

Typické **GO** signály:

- rozumieme problému a vieme ho modelovať
- riešenie sedí na naše skúsenosti a delivery model
- riziká vieme pomenovať a manažovať
- existuje reálna šanca, že vyhráme, nie len že sa zúčastníme

Ako skladáme víťazný tím

- Vlastník zákazky
- Bid Manager
- Solution Lead / Architekt
- Delivery Lead
- Pricer
- Security Lead
- T&T Lead (Projektový Manažér)
- Legal / právnik
- SME
-

Tím neskladáme podľa oddelení, ale podľa rolí potrebných na vyriešenie problému / prípravu ponuky.



4

Manažment Ponuky - Zákazky

Manažment zákazky

- Manažment (Governance) je riadený proces kontroly rozhodnutí počas celého životného cyklu potenciálnej zákazky.
- Definuje kto, kedy a na základe čoho:
 - schvaľuje pokračovanie projektu
 - potvrdzuje pripravenosť riešenia
 - preberá zodpovednosť za riziká
- Je súčasťou end-to-end procesu od:
 - kvalifikácie zákazky
 - cez návrh riešenia
 - až po zmluvu a jej podpis
- Rozhodnutia sa robia v kontrolných bodoch (reviews):
 - Qualification Review (QR)
 - Solution Review (SR)
 - Deal Review (DR)
 - Contract Certification Review (CCR)

Prečo sa Governance potencionálnej zákazky robí

- Governance chráni kvalitu riešenia, firmu aj projektový tím pred dôsledkami zlých rozhodnutí.
- Governance zabezpečuje, že:
 - riešenie je technicky realizovateľné
 - riziká sú identifikované a vedome akceptované
 - rozsah, cena a zmluva sú konzistentné
- Úroveň governance závisí od:
 - Veľkosti, typu a hodnoty projektu
- Povinné governance artefakty:
 - opis riešenia
 - architektonické rozhodnutia
 - identifikované riziká a predpoklady
 - schvaľovacie záznamy

Manažment ponuky = riadenie viacerých streamov

- **Ponuka nie je jeden dokument**
- Je to **súbeh viacerých streamov**, ktoré musia bežať spolu:
 - **Business / Sales** – vzťah so Zákazníkom, obchodná stratégia
 - **Bid / Pursuit** – smer, koordinácia, rozhodnutia
 - **Solution** – čo a ako dodáme (stratégia riešenia)
 - **Price** – koľko to stojí a prečo
 - **Legal / Risk** – čo si môžeme dovoliť
 - **Proposal** – ako to odkomunikujeme Zákazníkovi

Každý stream

- má mať **lídersku rolu**
- potrebuje **čas a prácnosť**
- má **vplyv na ostatné**

=> Ak zlyhá jeden stream, zlyhá ponuka

Manažment ponuky = orchestrácia, nie administratíva

Životný cyklus Zákazky

Zákazka má jasný životný cyklus

1. RFP / Q&A
2. Go / No-Go
3. Príprava riešenia
4. Cenotvorba / Optimalizácia
5. Schvaľovanie
6. Ponuka
7. Zmluva & Prechod do Delivery
8. Delivery

=> **Bez milníkov nie je kontrola**



Diskusia a prestávka



5

Tvorba a integrácia riešenia

Čo je „Riešenie (Solution)“: čo / ako / s kým / za koľko

- **Čo:** business problém a očakávaný výsledok
- **Ako:** architektúra + integrácia komponentov
- **S kým:** DXC, partneri, klient, subdodávateľia
- **Za koľko:** cost model, TCO, business case
- **Výstup:** riešenie ≠ produkt ≠ projekt, pridaná hodnota pre Zákazníka

Ako vzniká stratégia riešenia

Vstupy

- Požiadavky zákazníka (RFP / potreby / ciele)
- Kontext zákazníka (as-is, pain points, obmedzenia)
- Strategické ciele (business, IT, transformácia)
- Referenčné riešenia a skúsenosti
- Trhový kontext a konkurencia

Ako prebieha

Spoločná diskusia naprieč rolami (solution, delivery, sales)

Rozklad problému → návrh variant → rozhodnutia

Fokus na **trade-offs**, nie na detailné čísla

Iteratívne – výsledok sa spresňuje v čase

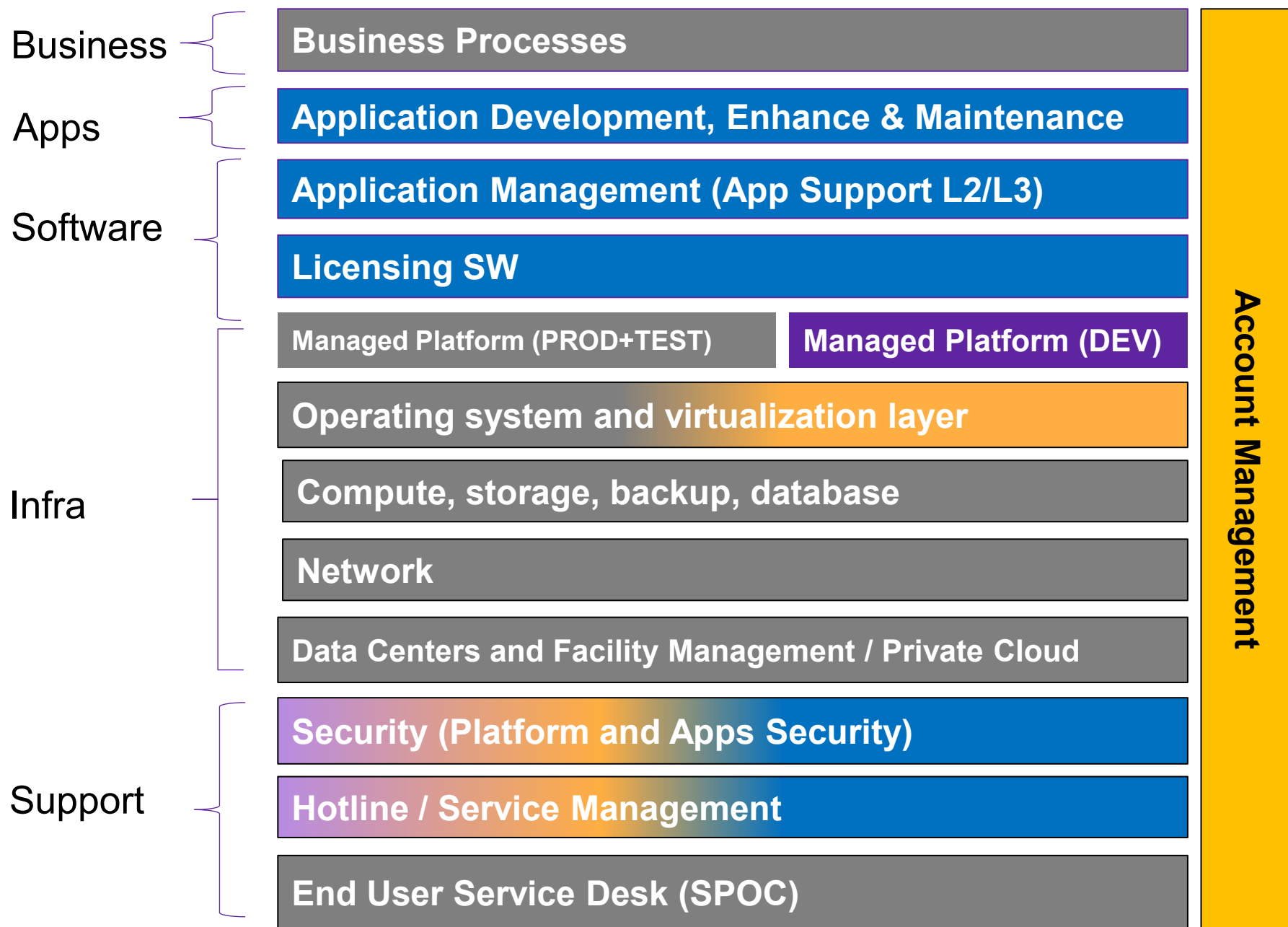
Výstupy workshopu

- Jasná **stratégia riešenia** („solution story“)
- Zvolený **smer architektúry a delivery model**
- 3 vektory (Hodnota pre biznis, Hodnota pre IT, Efektivita)
- Kľúčové **predpoklady, riziká a rozhodnutia**
- Vstup pre:
 - odhad nákladov
 - aplikáciu riešiteľských pák
 - cenotvorba a kontrolu nákladového modelu (Cost Clinic)

Architektonické pohľady

- **Biznis pohľad** – biznis procesy, schopnosti, hodnotové toky
- **Aplikačný pohľad** – aplikácie, funkcie, vlastníctvo
- **Infraštruktúrny pohľad** – compute, sieť, úložiská
- **Cloudový pohľad** – IaaS / PaaS / SaaS, zdieľané vs. dedikované
- **Bezpečnostný pohľad** – identita, dáta, compliance, hranice dôvery
- **Integračný pohľad** – rozhrania, dátové toky, kontrakty, zodpovednosti
- Pohľad na **riešenie Služieb** – príklad na ďalšom slide
- **Prevádzkový pohľad** – SLA, podpora, ticketing, monitoring, ITSM

Príklad – Prevádzkový pohľad – riešenie služieb



DXC scope:

- › Hotline support / Incident Management
- › Configuration of IT Platform (Managed Container Platforms based on K8S)
- › Monitoring
- › Application operations / Support L2&L3
- › Service Management (ITSM)
- › Application Development & Maintenance
- › Licensing of XXX SW above virtualization
- › Application Enhancement (Time & Labor = Rate Card)
- › Integrations from XXX with other systems (API)
- › IT Infrastructure operations (DEV env)

Outside DXC scope:

- › End Users Service Desk
- › Datacenter Services
- › IT Infrastructure operations (PROD +TEST env)
- › IT infrastructure assets (Server, Storage, Network, Backup)
- › Network Layer
- › Storage Layer
- › Hypervisor Layer (Docker / Kubernetes)
- › OS-subscriptions (Linux and Windows)
- › Database Layer



Integrácia: rozhrania > komponenty

- Komponenty sú **vymeniteľné**, rozhrania nie
- API / eventy / dávky = súlad medzi tímami
- Integrácia určuje **zodpovednosti**
- Najviac rizík je **medzi systémami**
- „Works standalone“ ≠ „works end-to-end“

Integrácia partnerov a subkontraktorov

- DXC ≠ automaticky vlastník všetkého
- Jasný **RACI** pre každé rozhranie
- Partner ≠ black box
- Eskalačná matica
- Zmluva musí odrážať architektúru

RACI je model, ktorý hovorí kto má akú rolu pri konkrétnej aktivite alebo rozhraní.

R – Responsible (vykonáva) - kto tú prácu reálne robí

A – Accountable (zodpovedá) - kto nesie konečnú zodpovednosť (iba jedno A)

C – Consulted (konzultovaný) - kto dáva vstupy / expertízu

I – Informed (informovaný) - kto musí vedieť, že sa niečo stalo

RACI nerieši hierarchiu, ale jasnosť zodpovedností.

Aktivita / Rozhranie	DXC	Partner (CRM)	Zákazník
Návrh API	A	R	C
Implementácia API	I	R	I
Integrácia / orchestration	R	C	I
End-to-end testovanie	A	R	C
Prevádzka po go-live	R	C	A

Delivery model ako súčasť architektúry

- Onsite / Nearshore / Offshore má **technický dopad**
- 24×7 support ovplyvňuje architektúru
- Security clearance, access, tooling
- Dostupnosť odborných zručností / kompetencií = architektonické obmedzenie
- Delivery model musí byť **navrhnutý**

Staffing plan

Kto

- roly a seniorita (L1/L2/L3, architekt)
- kľúčové a niche skills

Kedy

- Build / Transition / Run
- ramp-up a ramp-down v čase

Odkiaľ

- onshore / nearshore / offshore
- jazyk, časové pásma, regulácie

S akými zručnosťami

- skill availability na trhu
- štandardné vs. špecializované kompetencie

Z akého zdroja

- interní zamestnanci
- kontraktori
- nábor (čas + riziko)

V súlade s cenou a rizikami

- predpoklady a riziká ošetrené v SOW

DXC Príklad pre staffing plan

Aby to bolo doručiteľné

- Jasný scope & assumptions
- Realistická integrácia
- Zodpovednosti pokryté
- Tooling, procesy, governance
- Riziká identifikované a mitigované

Mini-ukážka: 3 najčastejšie integračné zlyhania

- Nejasný owner rozhrania
- Iné očakávania výkonu / SLA
- Integrácia riešená „na konci“

Čo je „Solution“ podklad pre schválenie doručiteľnosti

Solution ≠ prezentácia

- scope & architektúra
- delivery model
- cost model
- hlavné riziká a predpoklady

Checklist chráni firmu

- rozumieme scope/rozsahu
- náklady sú obhájiteľná
- riziká sú pomenované
- riešenie je doručiteľné

=> Nie byrokracia, ale poistka



6

Životný cyklus riešenia

Etapy riešenia IT

Typický model spolupráce a časová os



Zber informácií

- inventár aplikácií / infraštruktúry
- rozsah / zložitosť
- špeciálne požiadavky
- incidenty / export ticketov
- detaily prostredia



Analýza

- analýza aplikácií
- odhad pracnosti
- zložitosť prechodu
- prípady použitia automatizácie
- aplikácia produktivity



Ponuka

- business case
- medziročné úspory
- prístup / koncept riešenia
- prípady použitia automatizácie
- harmonogram

Projekt (T&T) vs. Prevádzka (Run)

- **Transition & Transformation (T&T)** – zmena stavu, migrácie, preberanie, stabilizácia
- **Run** – stabilná prevádzka, SLA, optimalizácia

- **T&T je dočasné,**
- **Run je dlhodobé,**

- **Iné ciele, metriky, iné riziká**
- **Architektúra musí fungovať v oboch fázach**

Transition: prechod do cieľového stavu

- Preberanie služieb, systémov a zodpovedností
- Migrácie dát, používateľov, rozhraní
- Nastavenie prevádzkových procesov (ITSM)
- Knowledge transfer a dokumentácia
- Stabilizácia po go-live
- Príklad (ďalší slide)

DXC príklad: Tranzícia

Transformation: zmena spôsobu fungovania

- Zmena procesov, nie len technológie
- Nové roly, nové zodpovednosti
- Zmena návykov a očakávaní
- Dočasný pokles výkonu je normálny
- Odpor k zmene je prirodzený

Prevádzka: SLA, stabilita, zlepšovanie, náklady

- Plnenie SLA a KPI
- Stabilita a dostupnosť služieb
- Incident, problem a change management
- Kontinuálne zlepšovanie
- Riadenie nákladov (Run > Build)

Zmluva & prechod od riešenia ku dodávke

- Prechod zo „stavby“ do „prevádzky“
- Zladenie zmluvy a reálneho riešenia
- Jasné prevzatie zodpovedností
- Nejasný handover = budúce konflikty
- Posledná šanca opraviť nesúlad



7

Cenotvorba

Prehľad ceny: ako vzniká cena riešenia

- **Základné náklady:** ľudia (FTE), infraštruktúra, softvér, služby, partneri
- **Riziká dodania (delivery risk):** nejasný rozsah, integrácia, závislosti, zmeny
- **Finančné riziká:** inflácia, výmenný kurz, dĺžka kontraktu, objemy
- Celkové náklady = Základné náklady + Delivery Risk + Finančné rizika
- **Marža:** primeraná riziku a typu zákazky (zvyčajne uvádzaná v %)
- **Cena = kombinácia všetkých týchto prvkov**

$$\text{Cena} = \text{Celkové náklady} / (1 - \text{marža})$$

Príklad:

Celk. náklady = 100 €

Marža = 30 %

Cena = $100 / (1 - 0,30)$

Cena = $100 / 0,70$

Cena $\approx 142,86$ €

Z čoho sa skladajú náklady na riešenie

- **Ľudia (FTE – Full-Time Equivalent)** – jednotlivé role pre fázú životného cyklu riešenia (T&T vs Run) a ich seniorita
- **Hardvér / Softvér** infraštruktúra, licencie, cloudové služby
- **Prevádzkové služby** podpora, monitoring, ITSM procesy
- **Partneri a subdodávatelia** fixné aj variabilné položky

- Náklady priamo vyplývajú z architektúry a modelu dodania

Náklady vyplývajú z delivery modelu

- **On-site (u Zákazníka)** - vyššia cena, dostupnosť na mieste, rýchla komunikácia
 - **On-shore (v krajine Zákazníka)** - vyššia cena, blízka dostupnosť, rýchla komunikácia
 - **Near-shore (blízke krajiny)** - kompromis medzi cenou a dostupnosťou
 - **Offshore (vzdialené centrá)** - škálovanie, nižšie jednotkové náklady
-
- Dopady na komunikáciu, kvalitu a stabilitu
 - Nie každá práca je vhodná na presun do zahraničia

Fixná cena a riziko

- **Fixná cena znamená obmedzenú flexibilitu**
- **Riziká v rozsahu, integrácii a závislostiach**
- **Rezerva (contingency) kryje neznáme**
- **Nie je to „voľný rozpočet“**
- **Bez rezervy sa riziko len presunie inde**

Rezerva nechráni zisk – chráni schopnosť dodať riešenie.

Kde vzniká únik nákladov

Príklady z praxe

- Nejasné integračné rozhrania
- Zmeny počas prevádzky bez preceňovania
- Nevhodný model podpory
- Slabé odovzdanie do prevádzky (knowledge transfer, handover)

Najdrahšie náklady sú tie, o ktorých sa na začiatku nehovorilo.

DXC Príklad pre model nákladov



8

Odhadovanie a Optimalizácia riešenia

Odhadovanie riešenia

Odhad je rozhodnutie, nie výpočet:

- Odhad vychádza z architektúry a rozsahu
- Neistota je prirodzená súčasť odhadu
- Presnosť rastie s kvalitou vstupov
- Zlý odhad = zdroj budúcich rizík
- Odhad musí byť obhájiteľný, nie „optimistický“

Vstupy do odhadu:

- Rozsah (Scope)
- Objemy a komplexita
- Riziká a obmedzenia
- Načasovanie a spôsob dodávky
- Predpoklady

Ako k odhadu pristupujeme:

- Porovnanie s podobnými riešeniami (benchmark)
- Rozklad na časti (top-down / bottom-up)
- Zapojenie skúsených rolí
- Práca s rozsahom neistoty
- Iteratívne spresňovanie

Odhadovanie riešenia: z čoho sa skladá výsledok

- Odhad riešenia sa skladá z **viacerých typov nákladov/práce**:
 - Každý typ má inú logiku výpočtu
 - Spolu tvoria obhájitelný výsledok

Náklady sú zväčša zoskupené do blokov:

- Jednorazové náklady napr. preberanie (Tranzícia), zmeny (Transformácia), konfigurácie, atď (typicky tvoria 8-20% celkových základných nákladov)
- Prevádzkové (opakujúce sa) napr. základná prevádzka a podpora, malé zmeny, a pod.
- Náklady na zmeny/projekty počas prevádzky napr. hromadné úpravy, rozvoj, change requesty
- Riadenie, governance, koordinácia (typicky tvoria 8-12% celkových základných nákladov)

Ako sa odhad počíta: objem × zložitosť × kontext

- **Objem práce** - tickety, zmeny, aplikácie, položky
- **Zložitosť** - Low / Medium / High
- **Kontext** - SLA, automatizácia, regulácia, integrácie

Ilustračná logika:

- Hodiny → FTE → náklady (Slovensko 140 hodín/mesiac, Egypt 160 hodín/mesiac)
- Automatizácia znižuje **hodiny**, nie zodpovednosť
- Obmedzenia zvyšujú **úsilie**, nie „len cenu“

- Odhad vždy stojí na **predpokladoch**
- Ak nie sú pomenované, **menia sa na riziká**

Rozdiel v praxi: odhady pre infraštruktúru vs. aplikácie

Infraštruktúra (jednoduchšie a presnejšie):

- vysoká miera **š**andardizácie
- existujú **KPI / normy / benchmarky**
- odhad je viac **mechanický**
- príklady:
 - správa serverov (počet serverov na FTE)
 - workplace / PC (počet zariadení na FTE)
 - service desk (tickety na FTE / mesiac)

Aplikácie (zložitejšie):

- každá aplikácia je do istej miery **unikát**
- veľký vplyv architektúry, integrácií a kvality kódu
- menej KPI, viac **expertného odhadu**
- silná závislosť od kontextu

Riešiteľské páky (Solution Levers)

Optimalizácia riešenia ako riadenie rovnováhy:

- Páky riešenia (solution levers) ovplyvňujú cenu aj riziko
- Nejde o šetrenie, ale o vedomé rozhodnutia
- Každá páka má technický aj biznis dopad
- Nesprávne použitie zvyšuje riziko
- Správne použitie zvyšuje udržateľnosť

Typy riešiteľských pák:

- **Páky rozsahu** - čo robíme, čo nerobíme, fázovanie
- **Architektonické páky** - zjednodušenie architektúry, menej integrácií, štandardy
- **Páky dodávkového modelu (Delivery model)** - onshore / nearshore / offshore mix, načasovanie rolí
- **Páky ľudí a zručností** - L1/L2/L3 rozdelenie, správna seniorita na správnu úlohu a s ohľadom na fázu projektu
- **Produktivita a automatizácia** - odstránenie manuálnej práce, self-service, runbooky, GenAI
- **Znovupoužiteľnosť** - opakované použitie komponentov, vzorov a nástrojov
- **Riadenie a governance** primeraná úroveň riadenia,
- **Riziká a predpoklady** - zníženie neistoty, jasné predpoklady, správna rezerva

Optimalizácia riešenia

Čo najviac ovplyvňuje pracnosť:

- Úsilie (effort) \neq počet ľudí
- Produktivita závisí od kontextu a zrelosti tímu
- Znovupoužiteľnosť znižuje pracnosť
- Štandardy a automatizácia pomáhajú
- Chaos a ad-hoc riešenia ju zvyšujú

Správna pyramída tímu

- Seniorné roly hore, exekúcia dole
- Nie všetko patrí off-shore
- Komplexné rozhodnutia bližšie k zákazníkovi
- Rutinné činnosti škálovať
- Nesprávna pyramída = skryté náklady

Obmedzenia modelu dodania

- Regulačné a zmluvné obmedzenia
- Krajiny, kde je možný off-shore
- Bezpečnostné a dátové obmedzenia
- Požiadavky zákazníka
- Súlad s internými pravidlami

Cost clinic / scrubbing: kontrola nákladového modelu

Aby bola náklady obhájitelné:

- Konzistentnosť odhadu a rozsahu
- Realistické predpoklady
- Aplikácia riešiteľský pák
- Súlad s delivery modelom
- Žiadne „magické čísla“
- Jasne pomenované riziká
- Explicitné predpoklady
- Čo je mimo rozsahu
- Čo musí dodať zákazník
- Čo sa nemení bez zmeny ceny

Príklad pre Cost Clinic workshop

Prečo

- byť „in to win“ a porovnať náklady s Cost-to-Win
- overiť, že všetky náklady sú zahrnuté (bez medzier/duplicít)
- zosúladiť objemy naprieč službami vrátane ich výkyvov
- vyjasniť aktivity a ich načasovanie

Ako prebieha

- iteratívne, nie jednorazovo; začať čo najskôr a nadviazať na Stratégiu riešenia
- najprv rýchly „high-level“ prehľad (≈ 60 min), potom „service-by-service“ do hĺbky
- odporúčanie: používať nástroj na vizualizáciu dát (napr. Power BI, atď)

Kto

- Lead Solution Architect zvolá, vedie a doručuje výsledok
- Riešiteľský tím: pripraví modely a vstupy
- Deal Analyst: benchmarky, market pricing, allowable cost
- Delivery stakeholderi: validujú realizovateľnosť a súlad so štandardami



Diskusia a prestávka



9

Manažment predpokladov, závislostí a rizík

Predpoklady vs. riziká

Čo je predpoklad (Assumption)

- Niečo, čo **musí platiť**, aby riešenie fungovalo
- Používame ho pri **odhade a návrhu riešenia**
- Príklady:
 - zákazník dodá vstupy včas
 - objemy ticketov neprekročia dohodnutý rozsah
 - integrácie sú dostupné a zdokumentované

Čo je riziko (Risk)

- Niečo, čo **sa môže pokaziť**
- Má **pravdepodobnosť a dopad**
- Príklady:
 - oneskorené vstupy
 - vyššia komplexita integrácií
 - nestabilné legacy prostredie

Prečo je rozdiel kritický

- Nepomenovaný predpoklad → **skryté riziko**
- Skryté riziko → **problém v delivery**
- Cost clinic kontroluje, či sa predpoklady **nezamieňajú za istotu**

Top assumptions / risks: ako sa zbierajú a udržiavajú

Kedy vznikajú

- počas **tvorby stratégie riešenia**
- pri odhade nákladov
- pri Cost clinic
- pri prechode do delivery

Ako sa zbierajú

- spoločná diskusia (solution, delivery, pricing)
- nie individuálne „v hlave architekta“
- zapisujú sa **explicitne**

Ako sa udržiavajú

- ako **živý zoznam**, nie jednorazový slide
- pravidelne revidované:
 - Pred odovzdaním ponuky
 - Pred podpisom Zmluvy
 - Pri zmene rozsahu
 - Pri prechode do delivery

Praktické pravidlo

- ak sa predpoklad zmení → **mení sa cena alebo rozsah (nemusí to platiť vždy)**
- ak sa riziko naplní → **aktivuje sa mitigácia**

Závislosti a integrácie ako zdroj rizika

Prečo sú integrácie rizikové

- nie sú plne pod našou kontrolou
- často závisia od:
 - tretích strán
 - zákazníka
 - legacy systémov

Typické rizikové závislosti

- dostupnosť a súčinnosť
- dostupnosť API a rozhraní
- kvalita a aktuálnosť dokumentácie
- synchronizácia zmien naprieč systémami
- release okná a schvaľovania

Čím viac integrácií, tým:

- vyššia zložitosť
- vyššie testovacie úsilie
- vyššie riziko eskalácií

Integrácie nie sú len technický detail – sú jedným z hlavných zdrojov rizika a nákladov.

Ošetrenie rizík: stratégie a zodpovednosti

Čo znamená „ošetriť riziko“

- Riziko **nemusíme vždy odstrániť**
- Musíme sa **vedome rozhodnúť**, ako s ním naložíme
- Každé riziko má mať:
 - zvolenú **stratégiu**
 - jasnú **zodpovednosť**

ACCEPT – akceptovať

- riziko je známe a akceptovateľné
- dopad je nízky alebo primeraný
- príklad:
 - mierne kolísanie objemu ticketov
- typicky:
 - riešené **v Solution**
 - sledované v delivery

AVOID – vyhnúť sa

- zmeníme riešenie tak, aby riziko **vôbec nevzniklo**
- príklad:
 - odstránenie zložitej integrácie
 - zúženie rozsahu
- typicky:
 - architektonické alebo scope rozhodnutie
 - riešené **v Solution**

MITIGATE – znížiť

- riziko zostáva, ale **znižujeme pravdepodobnosť alebo dopad**
- príklad:
 - dodatočné testovanie
 - Proaktívny monitoring
 - buffer v pláne
- typicky:
 - kombinácia **Solution + delivery opatrení**

TRANSFER – preniesť

- riziko **presúvame na inú stranu**
- najčastejšie:
 - na zákazníka
 - na partnera
- príklad:
 - závislosť na kvalite vstupov zákazníka
 - dostupnosť externého API
- **musí byť explicitne v SOW**
- inak riziko **ostáva u nás**

Ošetrenie rizík: počas prípravy ponuky vs Zmluva

Ošetrenia počas prípravy riešenia/ponuky:

- avoid / mitigate / accept
- architektonické a technické rozhodnutia
- zjednodušenie rozsahu
- použitie štandardov
- automatizácia
- návrh delivery modelu
- plánovanie kapacít a bufferov

V Zmluve / Scope of Work (SOW):

- transfer rizík
- povinnosti zákazníka
- SLA
- závislosti
- change management mechanizmy
- out-of-scope definície

Kritický princíp: prepojenie Solution ↔ SOW

Problémová situácia:

Riziko / predpoklad je známy, je pomenovaný v Solution, ale **nie je v SOW**

Výsledok:

Zmluvne neexistuje → v delivery je **plne na Poskytovateľa**



10

Od Ponuky cez Zmluvu ku Dielu

Ponuka: čo je „výstup“ pre Zákazníka

Čo klient reálne dostáva

- Opis riešenia
- Rozsah služieb a zodpovedností
- Cenu a cenový model
- Hlavné predpoklady a závislosti
- High-level delivery model

Čo Proposal NIE JE

- nie je technický detail
- nie je interný odhad
- nie je zmluva

Čo musí byť konzistentné: Riešenie ↔ Cena ↔ Riziká

Tri veci, ktoré musia sedieť

- **Riešenie**
 - rozsah, architektúra, delivery model
- **Cena**
 - effort, FTE, mix rolí, náklady
- **Riziká & predpoklady**
- čo musí platiť, aby cena dávala zmysel

Typické chyby

- lacná cena bez pokrytia rizík
- riešenie bohatšie než cena
- predpoklady len „v prezentácii“, nie v dokumentoch

SOW (opis rozsahu): kde sa ponuka mení na záväzok

Čo je SOW

- právne záväzný opis:
 - čo dodávame
 - ako
 - za akých podmienok

Typický obsah SOW

- rozsah služieb (in-scope / out-of-scope)
- SLA a KPI
- delivery model
- predpoklady a závislosti
- mechanizmus zmien (change management)
- zodpovednosti Zákazníka

Kritický princíp

- čo nie je v SOW, neexistuje
- čo nie je zmluvne ošetrené → riziko na našej strane

Zmluvné dokumenty: čo typicky existuje

Typická štruktúra

- **MSA (Master Service Agreement)**
 - rámcová zmluva
 - právne podmienky, platobné podmienky, zodpovednosti, liability, IP, GDPR, atď.
 - platí pre viacero SOW
- **SOW (Statement of Work)**
 - konkrétny rozsah a služby
 - viaže sa na MSA

Vzt'ah dokumentov

- MSA = „pravidlá hry“
- SOW = „konkrétny zápas“

Rozlišujeme

- **Zmluva o dielo (Project / Build)**
 - dodanie konkrétneho výsledku
 - rozsah, termíny, akceptácia
 - jednorazová alebo projektová cena typicky:
 - vývoj / implementácia
 - transformačný projekt
- **Zmluva o prevádzke služieb (Run / SLA)**
 - dlhodobá prevádzka a podpora
 - SLA, KPI, support model
 - pricing založený na objemoch / FTE / jednotkách
 - typicky:
 - aplikácie
 - infraštruktúra
 - service desk

Prechod do delivery: čo nesmie chýbať

Čo musí delivery dostať

- finálne **riešenie a rozsah**
- harmonogram riešenia (časovanie, milníky, projektový plan)
- **SOW + relevantné časti MSA**
- zoznam **predpokladov a rizík**
- dohodnuté SLA a KPI
- eskalačný a change process
- Staffing plan:
 - plán kapacít v čase, nie len „celkové FTE“
 - zdroj kapacít (existujúci zamestnanci / kontraktori / nábor)
 - rozdelenie: T&T (Projekt) / Run (Prevádzka)
 - rozdelenie: L1 / L2 / L3 / Architekt
 - onshore / nearshore / offshore mix
 - nábeh (ramp-up) a útlm (ramp-down)
 - kľúčové roly a ich zodpovednosti

Prečo je handover kritický

- delivery nečíta prezentácie
- delivery dodáva podľa zmluvy



Záver

Zhrnutie

- SW inžinierstvo začína skôr než kód (RFP, požiadavky, architektúra), je to systematická disciplína
- Riešenie je kombinácia ľudí, procesov, technológií a k nim prislúchajúcich nákladov
- Integrácia je ťažšia než vývoj, predstavuje najväčší zdroj rizík, nákladov a zlyhaní
- Zlé rozhodnutia v návrhu = drahé problémy v prevádzke, údržbe a podpore
- Odhad, cena, riziká a zmluva patria k sebe
- Delivery nezačína podpisom zmluvy
- PSI je nosný predmet, ktorý poskytuje mentálny základ reálne využiteľný v praxi IT riešení.