

# Beyond software

---

MARTIN TOMASEK



# Zadání

---

Vaše společnost byla oslovena malým lokálním auto servisem, který by rád rozšířil portfolio služeb.

A co dál? Ptejte se a zjistěte co vlastně potřebuje?



# Krok číslo 1 – Získání všech informací

---

1. Vypněte si při schůzce mobilní telefony
2. Vezměte na schůzku někoho, kdo je obeznámen z prostředí – kromě informací potřebujete i prodat
3. Navažte přátelskou atmosféru – dobré pití a jídlo je úspěšným ice-breakerem. Mimo to máte někoho z předchozího bodu.
4. Ptejte se na prostředí, na procesy, pracovní postupy, cíle, očekávání, budoucnost, zkušenosti s obdobnými projekty.
5. Detailně postupně rozeberte každý aspekt toho co vám zákazník řekne. Body si poznamenávejte, ale nedělejte na schůzce zápis jen highlity
6. Na úvodní schůzce není potřeba mít všechno do detailu analyzováno, to zvládnete na dalších.
7. Nedávejte od stolu žádný odhad. Nikdy!
8. Představte vaší firmu, zkušenosti a znalosti. Nezapomeňte zmínit sociální život. Zákazník vás obvykle nenajímá protože musí, ale protože chce a dobrý business man ví, že obchod a sociální provázanost je velmi důležitá.

# Zadání

---

Vaše společnost byla oslovena malým lokálním auto servisem, který by rád rozšířil portfolio služeb. Hlavním pilířem naší společnosti je spokojený zákazník, proto bychom rádi nabídli majiteli aut komplexní služby. Tím, že se soustředíme zejména na ojeté vozy, tak zde vnímáme jistý potenciál investice. Představujeme si to tak, že bychom nabízeli majitelům aplikace do mobilu a na PC, kde by viděli, co se na jejich vozu dělalo, fotodokumentaci, mohli by si naplánovat prohlídky vozů. Aplikace by jim prohlídku připomněla, provedla základní nacenění a podobně. Naši zaměstnanci by zase věděli, co za vozy budou servisovat a mohli by se na to řádně připravit. Do budoucna možná nějaké další aplikace pro Apple Car Play a Android. Naše společnost má aktuálně 3 mechaniky, 1 sekretářku a já zde reprezentuji majitele firmy a současně občas pomáhám na dílně. Mám tedy detailní přehled o všem, co je potřeba. Roční zisk firmy je cca 800tis. Chtěli bychom se dostat minimálně na 1,6 mil Kč. Mechanici jsou aktuálně poměrně vytížení, nicméně prostor pro další zakázky máme.

# Co je potřeba pro funkční software?

---

1. Potřeby zákazníka
2. Správná interpretace těchto potřeb
3. Kvalitní analýza, návrh řešení a kontrola dodávky
4. Správný návrh komponent s ohledem na jejich účel a životnost
5. Výběr správného provozního modelu
6. Stanovení správných procesů pro continuous delivery a integration
7. Dokumentování kódu, integrací, procesů a jejich významu a vlivu na systém
8. Správného výběru infrastrukturních komponent
9. A hlavně – Lidi (analytici, vývojáři, projektový manažeři, obchodníci, bezpečnostní konzultanti, architekti a infra experti, DB administrátoři, operátoři a pracovníci dohledu 24/7, správci aplikačních serverů, správci síťových prvků, správci energocentra, koordinační pracovníci,...)

# Krok číslo 2 – Získání technických požadavků

---

1. Rozšíření portfolia služeb o:
  1. Online objednávací a notifikační systém
  2. Evidenci zakázek a provázanost s účetními systémy
  3. Správu vozu pro klienty a pro zaměstnance
  4. Zpětnou vazbu od klientů
  5. Předběžnou kalkulaci ceny
2. Snížení administrativní zátěže
  1. Objednání je automatizováno – nemusí již řešit sekretářka potažmo šéf
  2. Správa vozu – převedení papírové dokumentace do digitální a současně větší spokojenost klienta
3. Snížení doby po kterou je vůz na dílně –
  1. je možné dopředu objednat díly => rychlejší dokončení zakázky => více zakázek a spokojenější klient

# Krok číslo 2 – Získání technických požadavků – Pozor něco nevyřešíte

---

## 1. Zvýšení zisku

1. Software v tomto případě negarantuje zvýšení zisku, ale může k tomu významně přispět

## 2. Zvýšení počtu zakázek

1. Software nezvýší počet zakázek, ale lze se zaměřit na sociální média, rebranding, webové stránky, nabízet související služby.

## 3. Snížení vytíženosti mechaniků

1. Zadavatel toto nepoptává a pokud by poptával tak to v takovém malém týmu nedává smysl.

# Víme co tedy děláme? – Jak by to podal náš CTO.

---

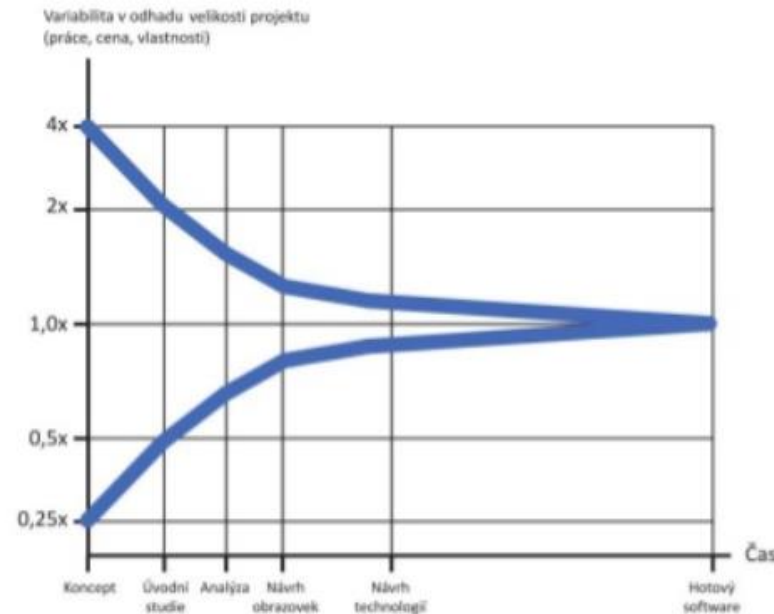
Pro klienta je nutné navrhnout komplexní aplikaci, která se bude primárně zabývat objednávání klientů a správou vozu. Kalkulace ceny bude nice-to-have, nicméně bychom chtěli, aby klient popsal, s čím má problém, pokud jede na nestandardní prohlídku. V konečném důsledku mu může zavolat sekretářka a doptat se ho na informace. Takže zakázka, respektive její průběh bude mít určitý životní cyklus. Dále má pro klienta smysl notifikační model pro zákazníky.

Fotodokumentace je též moc pěkný nápad, nicméně v rámci zisku firmy je to opravdu nice-to-have. Produkt chceme dostat co nejdříve na trh, tzn Time-to-market musí být v řádu týdnů. Zvažte model rolí a přihlašování. Soustředte se také na cenu provozu řešení a objem dat – dle mého názoru nebude nikterak velký. Vše ostatní vypusťte. Ještě zjistěte jeho stav na sociálních sítích, stav webu a jeho použitelnost a design vzhledem k nově vyvíjenému řešení.

A kolik to bude stát?



# Expertní odhad a kužel nejistoty



**Kužel nejistoty**  
reprezentuje  
statistickou chybu v  
odhadech, kterou  
udělali zkušení  
odhadovatelé na  
základě vstupních  
informací v  
jednotlivých fázích  
projektu.

# Krok číslo 3 – Use-case, popis chování systému, analýza řešení

---

1. Vytvořte dokument, kde popíšete základní chování systému
2. Určete aktéry systému a jejich činnosti
3. Navrhněte integrace mezi systémy.
4. Navrhněte DB vrstvu, počet tabulek, provázanost a popište složitou business logiku – je-li.
5. Navrhněte moduly, které potřebujete.
6. Promyslete objemy dat v budoucnu.
7. Promyslete financování analýzy a utvořte odhad ceny software (vývoj a provoz).

Poznámka pod čarou: Pro toto zadání nepotřebujete vytvářet 10 dokumentů. Stačí jeden, v kterém budou věci detailně popsány. Nezapomínejte, že dokument bude číst i zákazník a tak z něj musí pochopit co dostane a proč. Dokument mu nikdy posílejte, ale předejte mu ho osobně a v tištěné formě – digitální kopii až poté.

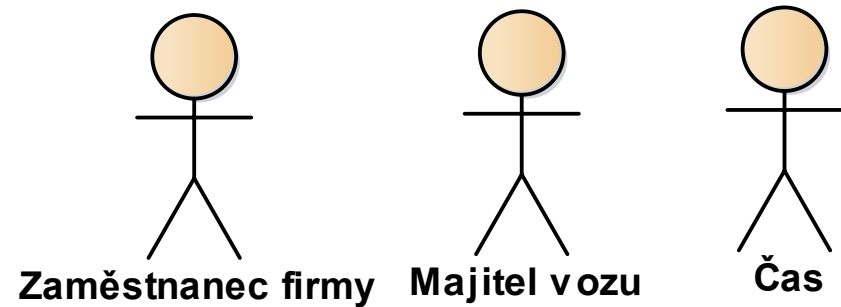
# Analýza

---

1. Rozeznáváme 2 role
  1. Klient (majitel vozu)
  2. Pracovník servisu
    1. Sekretářka?
    2. Mechanik?
    3. Majitel?
2. Klient (majitel vozu)
  1. Vytvoření, změna a odstranění objednání na servis
  2. Náhled na svůj vůz a zobrazení všech prohlídek
3. Zaměstnanec
  1. Správa zakázek servisu
  2. Náhled na vozy klientů

# Use-case

## uc Actors



Nezapomeňte provázat Aktory s use-case (zde není).

Zapomnělo se určitě na aktéra čas – kdo by posílal notifikace?

## uc Primary Use Cases

Hranice systému



# Technické řešení

---

1. Zvolíme servisně orientovanou architekturu. Proč?
  1. Více platforem
  2. Budoucí rozšiřitelnost
  3. Cena řešení a rychlé time-to-market
2. Autentizaci uživatelů budeme řešit pomocí OAuth protokolu
  1. Vlastní DB pro uživatele
  2. Možnost SSO z FB
3. Verze na PC bude JS klient, který provolává API.
4. Verze na mobil bude jaká?
  1. Nativní? iOS / Android
  2. Stejná jako pro PC (responsivní)?

# Technické řešení - moduly

---

1. Kolik tedy potřebujeme modulů?
  1. Modul pro vydávání tokenů
  2. Modul pro správu zakázek
  3. Modul pro správu uživatelů (nelze spojit s tokeny?)
  4. Modul pro správu vozů
2. Kolik potřebujeme FE
  1. Jeden – chceme ušetřit
  2. Mobilní verze použije webview – prozatím

# Technické řešení - infrastruktura

---

1. Kde budeme řešení hostovat
  1. Privátní server cca 2500kč/měs
  2. Cloud cca 300 kč/měs
2. Jak budeme chránit systém?
  1. DDOS?
  2. Ztráta a proniknutí dat do systému?
  3. Aplikační Gateway?
  4. Bezpečnostní monitoring?
  5. HA řešení?
  6. SLA?
  7. Monitoring

# Technické řešení – infrastruktura a aplikace

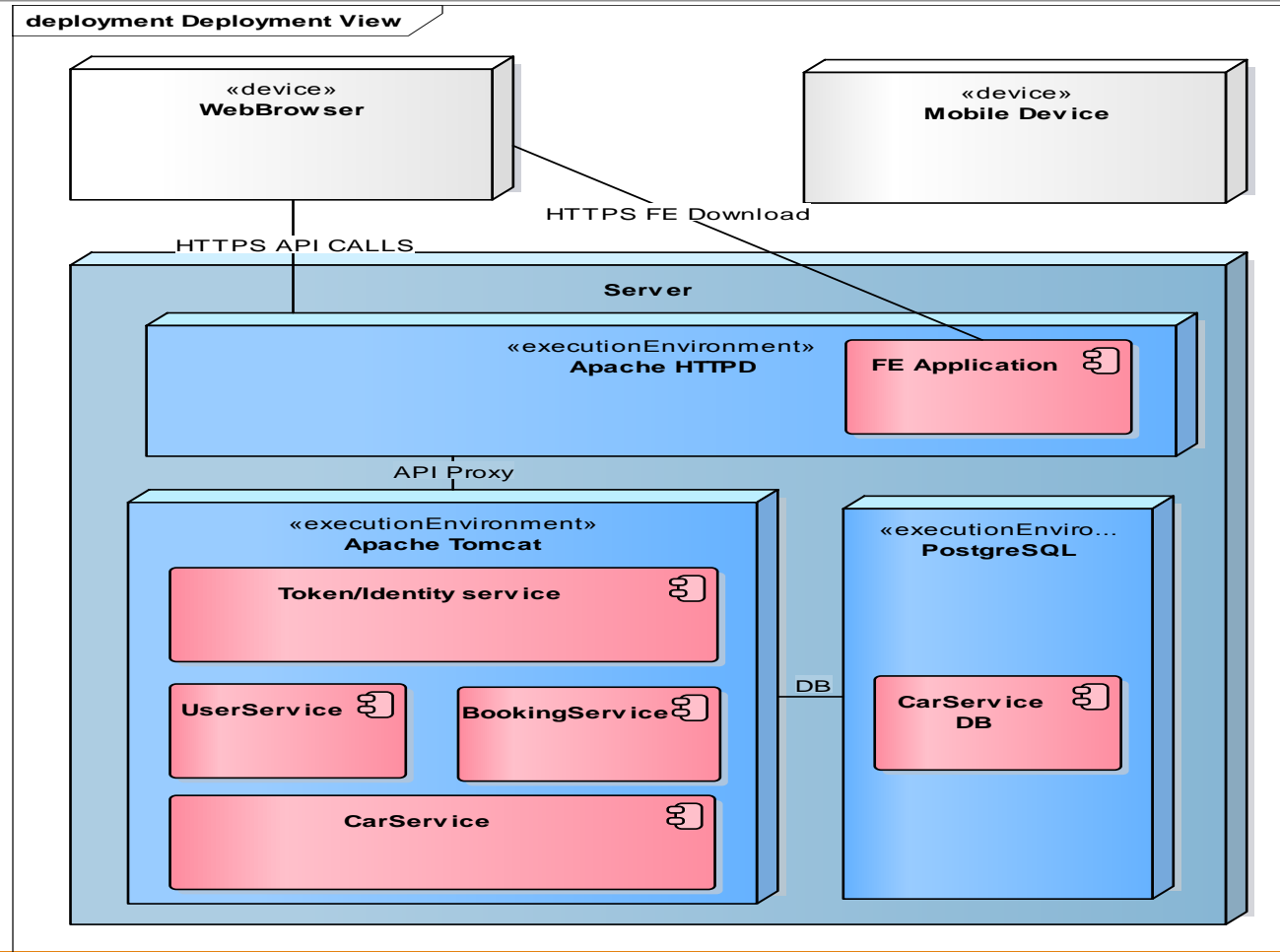
---

Vždy je závislá na velikosti projektu. Nebuďte megalomani 😊 V tomto případě postačí základní dogma:

1. Přístup na API chráněn SSL.
2. Jednotlivé moduly API běží na aplikačním serveru.
3. Server je vystaven za Apache HTTPD
4. API si vždy ověřuje token a role.
5. Chránění přes APIGW není nutné
6. Monitoring zajistíme externím dodavatelem (runscope apod)
7. SLA – není nutné mít 99.9%



# Technické řešení – infrastruktura a aplikace



# Technické řešení – DB návrh

---

1. Kolik budeme potřebovat tabulek?
  1. A) 5 - 10
  2. B) 10 - 15
  3. C) 15 - 20
  4. D) více než 20
2. Pro každý modul potřebujete specifické tabulky + vazební tabulky. V této fázi postačí konceptuální model. Počet DB určí náročnost software.
3. Jaké je tedy řešení?
  1. Uživatel (Jméno, Příjmení, Kontaktní informace)
  2. Vozy (VIN, SPZ, Značka)
  3. Servis (Vozidlo, Typ servisu, Příští návštěva, Co se provádělo, Datum servisu, Cena)
  4. Objednání (Vůz, Datum, Dílna, Mechanik, Odhadovaná doba)
  5. Tokeny (vydané tokeny, role (správa?))

# Technické řešení – DB návrh

---

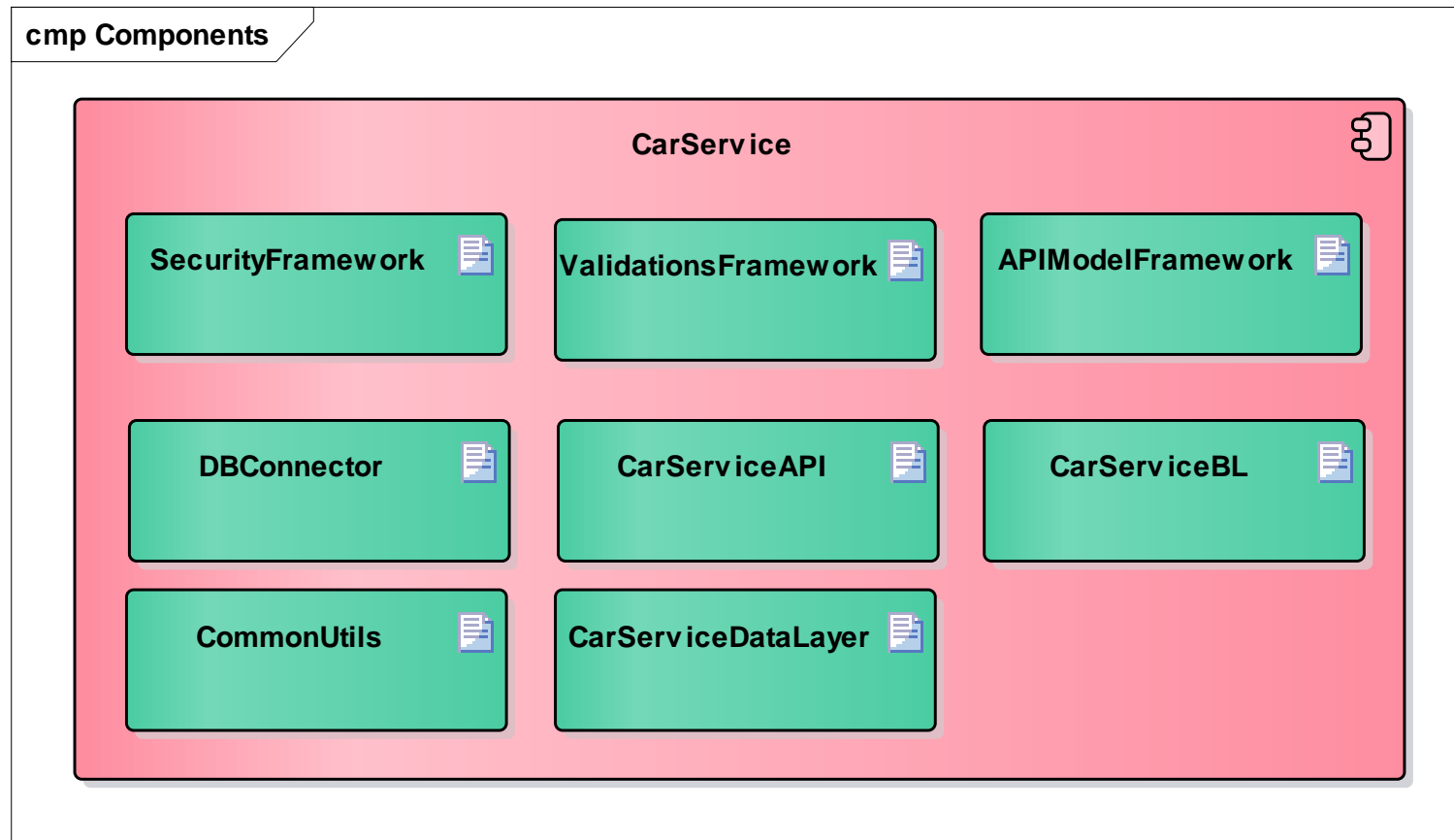
1. Pro realizaci potřebujeme celkem cca 10-15 Tabulek, včetně číselníkových – bez vazebních tabulek
2. Není potřeba extra logiky – jednoduché constrainty (pouze cizí klíče a validace na nenulovost)
3. Není nutné žádných složitých DB procedur, ani optimalizací.
4. Přístup více READ – více uživatelů bude číst než zapisovat
5. Pro každou tabulku potřebujete naimplementovat na API základní CRUD operace + řešit business validace + oprávněnost volání.

# Technické řešení – Architektura aplikace

---

1. Identifikujte společné prvky modulů a vytvořte je jako knihovny
  1. Ověřování tokenu, kontrola práv, získávání rolí
  2. Přístup do DB (DB connector)
  3. Validáční pravidla
2. Správně definujte build process (balíčkování – Gradle, Maven). Vytvořte různé profily (testovací X produkční)
3. Definujte logování aplikace, otáčení logů apod.
4. Všechny moduly ať používají stejnou technologii
5. Komunikace mezi moduly nemusí být v tomto případě nutně řízena přes APIGW nebo ESB
6. Nesdílejte věci mezi moduly, držte se REST principů

# Technické řešení – Architektura aplikace



# Krok číslo 4 – stanovení ceny a očekávání

---

1. Cena se stanovuje na základě odhady pracnosti SW (MDs)
  1. Počet DB tabulek
  2. Komplexita řešení
  3. Počet modulů
  4. Technologie
  5. Termín dodávky
  6. Výkonové nároky
  7. SLA
2. Fix Time / Fix Price, nebo Time-materiál
  1. V tomto případě Fix Time / Fix Price – zákazník potřebuje vědět kolik zaplatí
3. Kolik tedy?
  1. Obvykle počet MDs \* MD\_rate – Proč obvykle?

# Krok číslo 4 – stanovení ceny a očekávání

## Základní odhad 111MDs

Položka		MDs							
Analýza		5							
Návrh		3							
Vytvoření tech stacku		5							
Vytvoření modulu Tokenu									
	DB modelování	2							
	Integrace na DB + logika	5							
	Modelace API	2							
	Vytvoření API	3							
	Testování	2							
Vytvoření modulu Uživatelů									
	DB modelování	2							
	Integrace na DB + logika	1							
	Modelace API	2							
	Vytvoření API	1							
	Testování	2							
Vytvoření modulu vozů									
	DB modelování	3							

# Krok číslo 4 – stanovení ceny a očekávání

---

1. Vývoj 111MDs (potřebujete FE a BE developera)
2. Podpora řešení (potřebujete FE a BE developera / infrastrukturistu), minimálně však cca 2-8 hodin týdně
3. Provozní náklady ( cloud měsíčně cca 400kč, server 2500kč, server u zákazníka (nesmysl))
4. Náklady na technologický dluh
5. Nepočítáme náklady na analýzu na sociálních sítích.

Podrženo sečteno software bude stát vyrobít minimálně **500tis** + jeho provoz bude ročně vyžadovat cca **280tis**. Počítaná MD RATE **4800Kč**.



# Ostatní kroky

---

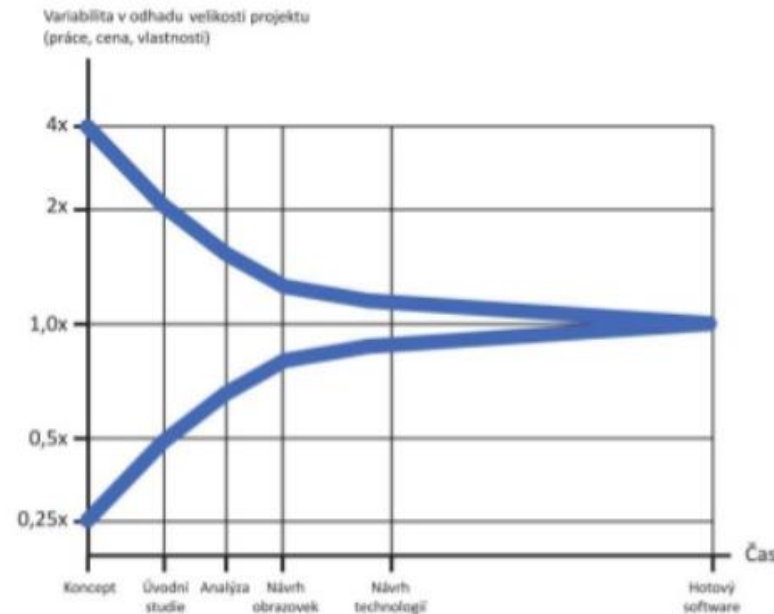
1. Začínáme vývoj
  1. Plánování, rozpad tasků
  2. Vytvoření vhodných nástrojů k buildu a správně zdrojových kódů
2. Budování testovacího prostředí
  1. Otestování budoucí integraci na produkci. Vytvoření identického prostředí.
  2. Konfigurace Serveru, Apache HTTPD, Apache Tomcat, DB
3. Budování produkčního prostředí
4. Ověření konfiguračního managementu, definice SLA
5. Dokumentace
6. Akceptace a akceptační kritéria
7. Zaplatit vývojáře
8. Neustálé meetingy se zadavatelem
9. Pokrýt náklady a vytvořit zisk – Povedlo se? Kolik jste vidělali?
10. Po projektu alokovat vývojáře na další projekt

# Na co jsme zapomněli?

---

1. Legislativní požadavky
  1. Souhlas se zpracováním osobních údajů
  2. Právo být zapomenut (GDPR)
2. Onboarding klientů
  1. Jedná se o nový software – jak do něj dostaneme lidi? Registrace?
3. Offboarding klientů
  1. I offboarding může vést k tomu, že se klient vrátí 😊
4. Síťová infrastruktura
  1. V takto malém projektu není potřeba – vyřešíme správným hostingem
5. Nice-to-have věci 😊

# Expertní odhad a kužel nejistoty



**Kužel nejistoty** reprezentuje statistickou chybu v odhadech, kterou udělali zkušení odhadovatelé na základě vstupních informací v jednotlivých fázích projektu.

# Dotazy a diskuze

---