service Transition

Obsah

[service Transition 1](#_Toc177455799)

[úvod 2](#_Toc177455800)

[Primary Breakdown – základné rozdelenie 2](#_Toc177455801)

[Change Management 3](#_Toc177455802)

[Change Management – úvod 3](#_Toc177455803)

[Change Management – Seven R‘s 3](#_Toc177455804)

[Change Management – RfC 4](#_Toc177455805)

[Change Management – Change Evaluation 5](#_Toc177455806)

[Change Management – Change types 5](#_Toc177455807)

[Change Management – CAB 7](#_Toc177455808)

[Service Asset and Configuration Management 8](#_Toc177455809)

[SACM – CMDB 9](#_Toc177455810)

[SACM – PROCESY 9](#_Toc177455811)

[Knowledge Management 10](#_Toc177455812)

[Knowledge Management - SKMS 10](#_Toc177455813)

[Release and Deployment Management 12](#_Toc177455814)

[R&DM – základné pojmy 12](#_Toc177455815)

[R&DM – fázy 12](#_Toc177455816)

[Application Development and Customization 13](#_Toc177455817)

[Service Validation and Testing 14](#_Toc177455818)

[Service Validation and Testing – Test Models 15](#_Toc177455819)

[Project Management – Transition Planning and Support 16](#_Toc177455820)

[Celkový obraz 17](#_Toc177455821)

[Zhrnutie 17](#_Toc177455822)

úvod

**Service Transition** (Prechod služby) je časť životného cyklu služby, kde je služba alebo ktorákoľvek z jej častí zostavená, upravená a nasadená. Pre praktické účely sa procesy, ktorými sa spravujú znalosti - **knowledge**, najmä technické znalosti a technické záznamy, tiež považujú za súčasť Service Transtition.

Predtým, ako sa ponoríme hlbšie do Service Transition, pozrime sa, čo sa so službou doteraz stalo.

Naša fiktívna služba začala svoju existenciu ako koncept s určitou **stratégiou** – bola koncipovaná ako spôsob uspokojovania potrieb zákazníkov a generovania príjmov pre našu organizáciu.

Tieto obchodné požiadavky boli následne postúpené do **Service Design**, ktorý tieto požiadavky prevzal a vypracoval **blueprint** – plán, podrobný popis špecifických technológií – ako je hardvér, softvér a prepojovacie káble, ako aj podrobný popis toho, aký druh informácií sa má uchovávať a kde, kto k nemu bude mať prístup alebo ho bude môcť zmeniť, aké budú úlohy zúčastnených a ďalšie.

**Zatiaľ sa však nič nepostavilo - všetko je „len“ na papieri**, alebo skôr v elektronických dokumentoch. Preto je cieľom Service Transition prijať tieto podrobné popisy a skutočne vybudovať, vyvinúť a nasadiť technológiu, architektúru a službu.

Ako sekundárny cieľ chceme, aby budovanie, nasadenie a zmeny služieb prebiehali koordinovane a efektívne.

Primary Breakdown – základné rozdelenie

Service Transition je často **opätovne navštevovaná** fáza životného cyklu služby. Niektoré procesy v rámci nej sa používajú pri **každej návšteve**, iné procesy vstupujú do hry len vtedy, keď má daný prechod (tranzícia) určité vlastnosti. Skôr než pôjdeme hlbšie, stručne si predstavíme procesy.

Základným procesom, ktorý sa vždy používa, je **Change Management** (riadenie zmien). V podstate vždy, keď sa má čokoľvek zmeniť – a to platí aj pre situáciu, keď je počiatočný stav „nič“ – Change Management dbá na to, aby všetky činnosti potrebné k tejto zmene boli vykonané koordinovane a profesionálne.

Aby to fungovalo, potrebujeme spoľahlivé a aktuálne informácie o aktuálnom stave vecí. Potrebujeme vedieť, kde sa aktíva (assets) nachádzajú, ako sú prepojené a nastavené a na akú službu alebo zákazníka sa používajú. Tieto informácie spravuje proces s veľmi komplikovaným názvom – **Service Asset And Configuration Management.**

*To sú dva procesy, ktoré sú absolútne minimum a používajú sa zakaždým.*

Niekedy je zmena, ktorú zamýšľame urobiť, taká veľká, čo sa týka rozsahu alebo dopadu (**scope or impact**), že musíme podniknúť ďalšie kroky, aby sme zabezpečili, že bude fungovať. V takýchto prípadoch používame **Change Evaluation** (hodnotenie zmien). Podobne, keď veci zmeníme vo veľkom, môže byť potrebné dodatočné overenie funkčnosti. Na tento účel ITIL ponúka proces **Service Validation and Testing** (validácie a testovania služieb) – ako už názov napovedá, kontrolujeme, či **celá** služba funguje tak, ako chceme, a nielen jej technické aspekty.

Zmeny, ktoré máme v úmysle vykonať, si niekedy vyžadujú úpravy alebo vývoj softvéru alebo databáz. **Application Development** (Vývoj aplikácií) je proces, ktorý riadi ich vytváranie, a pretože pred uvedením do prevádzky musíme funkčnosť otestovať, toto testovanie a súvisiace témy má na starosti ďalší proces s názvom **Release And Deployment Management** (Riadenie uvoľnení a nasadení).

Okolo všetkých týchto vysoko technických procesov sú ďalšie dva procesy, ktoré tiež patria k Service Transition:

**Knowledge Management**, ktorý zabezpečuje, že znalosti sa zbierajú, uchovávajú a sprístupňujú na opätovné použitie, a **Project Management**, ktorý zabezpečuje, že obzvlášť veľké zmeny sa uskutočňujú v súlade s ich odhadovaným rozpočtom, časovými a kvalitatívnymi obmedzeniami.

Na odlíšenie projektového manažmentu v Service Transition od iných typov projektového manažmentu sa niekedy namiesto neho používa názov **Transition Planning and Support** (Plánovanie a podpora prechodu).

Change Management

Change Management – úvod

**Úlohou Change Managementu je riadiť životný cyklus všetkých zmien.**

**Cieľom je zabezpečiť, aby sa uskutočnili prospešné zmeny s minimálnym narušením IT služieb.**

Aby sme pochopili význam tohto, musíme pochopiť, že IT služby poskytované spoločnosťami, ako je tá naša, sú podstatne zložitejšie ako to, čo ľudia používajú vo svojich domácnostiach. Doma bežne odpájame a pripájame externé úložné zariadenia, pretože si plne uvedomujeme, či počítač práve niekto používa alebo nie.

Keď chceme počítač premiestniť na iné miesto, odpojíme ho od elektriny aj siete, premiestnime ho a potom znova pripojíme.

Keď chceme použiť tlačiareň alebo skener, zapneme ich a keď ich už nechceme používať, vypneme ich, pretože je zrejmé, či ich niekto používa alebo nie.

Ale rozsiahle IT služby majú architektúry, ktoré pokrývajú viaceré budovy, viaceré mestá alebo dokonca viaceré kontinenty. Existujú desiatky, stovky alebo tisíce zložitých počítačových systémov, úložných systémov a databázových systémov, ktoré musia byť všetky navzájom prepojené, ale zároveň od seba tienené. Blok dát, ktorý sa posiela od jedného používateľa k druhému, ktorý sa nachádza niekoľko metrov od seba, môže cestovať po celom svete, môže byť počas cesty zašifrovaný a dešifrovaný a uložený na neskoršiu analýzu alebo na zálohovanie.

V takomto prostredí sa niečo vypnúť, odpojiť od siete alebo presunúť na iné miesto nezaobíde bez riadneho plánovania a informovania zodpovedných osôb.

Ako príklad mimo IT si predstavte bežnú situáciu v reštauráciách, keď si väčšia skupina zákazníkov chce spojiť dva alebo viac stolov, aby mohli sedieť spolu. Zdvorilí hostia by sa opýtali personálu, či je to v poriadku, a možno by im personál navrhol stoly ďalej od kuchyne, kadiaľ nemusia tak často chodiť. Žiaľ, niektorí hostia **robia takéto zmeny bez toho, aby si pýtali povolenie**, a personál sa potom musí pohybovať okolo ich „dvojstola“, čo môže spôsobiť najrôznejšie komplikácie.

Change Management – Seven R‘s

Change Management proces sa začína aktivitou s názvom **„Raising a change“** (Vyvolanie zmeny).

Vždy, keď dôjde k zmene - Change, je potrebné ju správne opísať a je potrebné zodpovedať 7 kľúčových otázok, známych ako sedem R.

Prvá otázka je „Kto vyvolal **(raised the change)** zmenu“ Odpoveď na túto otázku je známa ako **requestor** (žiadateľ). Táto otázka zabraňuje až príliš častej situácii, keď ľudia vykonajú aktivitu, ktorá skončí zle, a na otázku „Prečo ste to urobili?“ odpovedajú „Niekto mi to povedal“ bez toho, aby vedeli, kto ten niekto bol.

Ďalšia otázka je „Aký je **reason** (dôvod) zmeny“. Ak žiadateľ nemôže odôvodniť change, nemala by sa žiadna zmena vykonať.

"Aký je požadovaný **return** (prínos) zmeny?" je otázka, ktorá si vyžaduje konkrétny popis želanej koncovej situácie. Vedieť, ako by mali systémy vyzerať po zmene, umožňuje kontrolovať, či bola zmena úspešná.

"Aké sú **risks** (riziká) zmeny?" je otázka, ktorá nám umožňuje vidieť, čo sa môže pokaziť. O rizikách sme už podrobne hovorili. Tu chceme vedieť, aká je pravdepodobnosť, že zmena zlyhá, a aké by mohli byť dôvody jej zlyhania. To nám tiež umožní neskôr kategorizovať zmenu z hľadiska rizika.

Piata otázka znie: Aké **resources** (zdroje) sú potrebné na zmenu? Bez toho, aby ste mali túto informáciu, change sa pravdepodobne zasekne uprostred realizácie, pretože niečo dôležité bude chýbať – možno skrutkovač, možno druhý pár rúk, možno povolenie vstúpiť na zabezpečené miesto.

"Kto je **responsible** (zodpovedný) za zostavenie, testovanie a implementáciu?" je ďalšia otázka. Odpoveď objasní úlohy a zodpovednosti počas celého procesu.

A nakoniec: "Aké **relationships** (vzťahy) existujú medzi touto a inými zmenami?" je otázka, ktorá má za cieľ zabrániť kolíziám s inými zmenami. V tom istom systéme jednoducho nemôžeme vykonávať viacero zmien súčasne. A ak sa má vykonať séria zmien, potom musíme vedieť, ktoré zmeny sú súčasťou tejto série. Každá zmena v tejto sekvencii bude musieť počkať na dokončenie predchádzajúcej zmeny a overiť jej úspešnosť.

Skúste si zapamätať týchto sedem R, pretože tvoria základ dobrého Change managementu.

Change Management – RfC

Change management môže byť spustený niekoľkými rôznymi spúšťačmi. Niektoré z nich budú popísané v kapitole Service Operation (Prevádzka služby), ale zatiaľ sa budeme baviť len o najuniverzálnejšom spúšťači.

Tento spúšťač sa nazýva **Request for Change**, alebo **R-f-C**, a je to presne to, čo názov napovedá - ide o písomnú požiadavku na vykonanie určitej konkrétnej zmeny. Request for Change (žiadosť o zmenu) pochádza od zákazníka alebo používateľa a requestor (žiadateľ) by mal zvyčajne uviesť dôvod takejto žiadosti. Keďže sme sa už predtým dotkli scenára reštauračných stolov, požiadavka môže znieť: „Mohli by ste naše stoly posunúť k sebe? Sme väčšia skupina a radi by sme sedeli spolu."

Pomáha, ak sú k dispozícii šablóny pre R-f-C. Tieto **RfC templates** môžu previesť zákazníka alebo používateľa procesom správneho vyplnenia všetkých potrebných údajov. Tu je veľkou výhodou skutočnosť, že poskytujeme IT služby, pretože môžeme bezpečne predpokladať, že výpočtová technika je žiadateľom dostupná. To nám umožňuje implementovať takéto šablóny vo forme webových formulárov alebo decision trees (rozhodovacích stromov) s hypertextovými odkazmi a vykonávať všetky druhy kontrol validácie údajov. Môžeme napríklad potrebovať, aby requestor poskytol e-mailovú adresu alebo identifikáciu systému a pomocou moderných technológií môžeme okamžite overiť, či sú poskytnuté informácie platné – môžeme zistiť, či systém existuje alebo či e-mailová adresa funguje.

Postupom času sa organizácia stáva vysoko kvalifikovanou pri vykonávaní určitých zmien. V našom scenári reštaurácie je v podstate každý čašník a čašníčka už vyškolený na výmenu obrusov. Každý čašník vie, ako v letnej sezóne premiestniť stoličky na terasu a zase späť, alebo ako večer naaranžovať sviečky na stoly a zapáliť ich.

Pre takéto zmeny vieme definovať **Change Model** (vzor zmeny) – presný štandard, ako ich vykonať, vrátane presného načasovania a dostupných zdrojov. Zmeny vykonané podľa modelu zmien sa zvyčajne nazývajú **Standard changes** (štandardné zmeny) a vo všeobecnosti vyžadujú len minimálne schválenia. Kvôli tomuto spojeniu medzi **Change Models** a **Standard Changes** sa tieto zmeny niekedy nazývajú Model Changes (modelové zmeny).

Keď sa implementuje change, môže si vyžadovať vypnutie alebo odpojenie príslušného systému. Ak sa systém stane nedostupným počas implementácie zmeny, nazýva sa to **downtime** (odstávka). Pred schválením zmeny, ak si vyžaduje odstávka, bude musieť zodpovedná osoba zvyčajne odhadnúť trvanie odstávky a zodpovedajúci výpadok služby. Toto sa nazýva **Projected Service Outage** (plánovaný výpadok služby) a musí sa zvážiť vo vzťahu k dohodnutým Service Levels a dostupnosti služieb.

Každá zmena musí byť zaznamenaná v správnom systéme, ktorý potom obsahuje **Change Records** (Záznamy zmien). Change Record je súbor všetkých informácií o konkrétnej zmene a zvyčajne sa aktualizuje vždy, keď sú k dispozícii ďalšie informácie. Záznam môže na začiatku obsahovať kópiu Request for Change (žiadosti o zmenu), ktorá bola použitá na jej vyžiadanie. Keď sa zmena vyhodnotí, pridelí realizátorom (implementers) a naplánuje sa na vykonanie v konkrétnom čase, tieto informácie sa pridajú do Change Record. Pre praktické účely má každá change aj **status**, ktorý označuje, ako riadenie tejto zmeny postupuje – tieto statusy môžu byť napríklad „Open“, „Accepted“, „Scheduled“, „In progress“ a „Closed“. Výhodou takéhoto stavu je možnosť filtrovať databázu zmien a vykonávať kontroly kvality alebo analýzy konkrétnej podskupiny.

Change Management – Change Evaluation

Change môže byť niečo také jednoduché, ako je presun počítača používateľa z jednej kancelárie do druhej, alebo niečo také zložité, ako je migrácia tisícky serverov z dátového centra v Nemecku do dátového centra vo Francúzsku. Je zrejmé, že zložitosť činností, ktoré sa musia v každom z týchto prípadov uskutočniť, je odlišná a sú s nimi spojené rôzne riziká. ITIL definuje proces **Change Evaluation** (hodnotenia zmien), ktorý nám poskytuje dôležité informácie o zložitosti a riziku konkrétnej zmeny. Výsledky takéhoto hodnotenia sú prezentované v **Change Evaluation Report** (správe o hodnotení zmien). Táto správa sa používa pri viacerých príležitostiach v rámci životného cyklu zmeny.

Predstavte si situáciu, že sa reštaurácia pripravuje na špeciálnu párty, ktorá si vyžaduje veľkú reorganizáciu – treba presťahovať nábytok, meniť dekorácie a podobne. Change Evaluation by zohľadňovalo veci ako „Ako kvalifikovaní sú zamestnanci – robili už takéto veci? a "Sme si istí, že všetok nábytok sa zmestí cez dvere a chodby?" Ak odpoveď na prvú otázku bola „Vôbec nie som kvalifikovaný“, evaluation report môže zmenu kategorizovať ako **high risk change** (vysokorizikovú zmenu).

To, ako spoločnosť pristúpi k zmenám s vysokým rizikom, závisí od ich obchodnej stratégie: Niektoré spoločnosti majú prísnu politiku rozdeliť takéto zmeny na menšie zmeny, iné môžu byť ochotné podstúpiť takéto riziko a zmierniť ho prostredníctvom poistenia.

V prípade reštaurácie by jedným z možných prístupov bolo vykonať test – skúsiť vykonať zmenu niekoľko dní vopred, zistiť prípadné problémy a potom buď prísť s riešeniami, alebo vykonať úpravy v plánovanej zmene, ktoré by tieto problémy odstránili.

Jednou špecifickou kategóriou zmien sú **Major Changes** (Zásadné/veľké zmeny). Major Changes sú zmeny, ktoré zavádzajú nové služby alebo robia podstatné zmeny existujúcich služieb. Veci ako premiestnenia alebo upgrady všetkých systémov na novšiu verziu alebo prechod z jednej technológie na druhú, to všetko sú príklady Major Changes.

Change Management – Change types

Môžeme povedať, že Change Management proces je dôkladne plánovaný proces: Všetko by sa malo robiť po premyslení, zvážení čo najväčšieho počtu faktorov, zhodnotení rizík, skontrolovaní možných konfliktov v plánovaní, informovaní všetkých vopred atď. To určite platí pre prípady, keď je zmena vykonaná ako Standard Change, alebo ako súčasť nejakého väčšieho projektu.

Zmeny však môžu vyvolať aj mimoriadne udalosti. Ak sa počítačový systém poškodí v dôsledku technickej poruchy alebo ak prestane reagovať v dôsledku vírusovej infekcie alebo útoku, musíme ako poskytovateľ služieb zvyčajne rýchlo reagovať, aby sme zabránili ďalšiemu poškodeniu a obnovili normálnu prevádzku. Ak si náš zásah vyžaduje zmenu niečoho, napríklad výmenu chybného hardvéru, potom bude potrebná zmena. Takáto zmena sa nazýva **Emergency Change** (núdzová/mimoriadna zmena).

Poďme ešte raz do našej reštaurácie: Jednému z hostí omylom spadla sviečka na obrus a ten sa vznietil. Našťastie čašníčka, ktorá stála neďaleko, rýchlo zareagovala a vyliala na stôl vodu, čím oheň uhasila skôr, ako mohol napáchať vážne škody, no aj tak je v obruse obhorená diera a zákazníci sú trochu otrasení. Je jasné, že ich bude treba premiestniť k novému stolu a rýchlo vymeniť poškodený obrus.

Za normálnych okolností sa proces výmeny obrusov vykonáva iba vtedy, keď sú všetky stoly prázdne, ale keďže na to nemôžeme čakať, vykoná sa Emergency Change.

Jeden z našich zamestnancov príde so servírovacím vozíkom, položí naň všetky jedlá a riad a odnesie ho k prázdnemu stolu, ku ktorému budú zákazníci premiestnení. Potom je možné poškodený obrus opatrne vymeniť - tým sa zabezpečí, že ostatní hostia nebudú rušení. Nakoniec sa na stôl položí nový obrus, podlaha sa vysuší a pozametá a stoličky zasunú naspäť.

Emergency Changes sa, samozrejme, nedajú posudzovať tak striktne ako bežné zmeny, pretože sme nemali možnosť všetko poriadne naplánovať a vyhodnotiť a pohybovali sme sa v situácii, keď už bola služba aspoň do určitej miery narušená. Naším cieľom je však realizovať Emergency Changes tak, aby nenarušili prevádzku služby viac, ako je nevyhnutné.

Za zmienku tiež stojí, že moderný hardvér je vybavený čoraz sofistikovanejšími nástrojmi na samodiagnostiku, niekedy nazývanými aj nástroje SMART, ktorých cieľom je vopred signalizovať, že zariadenie môže zlyhať. To je veľmi výhodné pre poskytovateľov **high-availability** (vysoko dostupných služieb) – služieb, ktoré zákazníci skutočne potrebujú mať neustále v prevádzke. Vďaka autodiagnostickým nástrojom, redundancii a veciam ako softvérové ​​a dátové cloudy sme schopní poskytovať služby, ktoré zdanlivo nikdy nezlyhajú.

**Change types podľa ITILu (vybrané, len pre doplnenie)**

V ITIL je Change Management kategorizovaný do rôznych typov na základe povahy a úrovne rizika spojeného so zmenou a sú to:

**1. Standard Change**

Štandardné zmeny konkrétne označujú nízkorizikové, rutinné zmeny, ktoré boli vopred schválené (napr. reset hesla, aktualizácie softvéru). Vždy sa riadia vopred definovaným postupom (tzv. Change Modelom).

**2. Normal Change**

Ide o zmeny, ktoré sa nepovažujú za štandardné zmeny, no napriek tomu majú nízku mieru rizika. Môže ísť o pridávanie nových používateľov, nového hardvéru alebo softvéru. Pred implementáciou si vyžadujú autorizáciu a schválenie. Na implementáciu tohto typu zmeny sa zvyčajne vyžaduje podrobný Request for Change.

**3. Emergency Change**

Ide o zmeny, ktoré je potrebné zaviesť okamžite, aby sa vyriešil naliehavý problém. Môže ísť o obnovenie služby po zlyhaní alebo použitie critical security patch (kritickej bezpečnostnej záplaty). Vyžadujú autorizáciu a schválenie, ale proces je zrýchlený, aby sa minimalizoval čas potrebný na implementáciu zmeny. V tomto prípade môže byť Request for Change spracovaný po vykonaní tejto zmeny.

**4. Major Change**

Ide o zmeny, ktoré majú výrazný vplyv na IT infraštruktúru, služby či používateľov a vyššiu mieru rizika. Príklady zahŕňajú implementáciu nových systémov alebo veľké aktualizácie. Pred ich implementáciou si vyžadujú podrobné posúdenie, autorizáciu a schválenie.

**5. Urgent (Naliehavá) Change**

Tento typ zmeny je podobný Emergency Change, ale nemusí nevyhnutne súvisieť s riešením incidentu. Môžu sa vyžadovať, aby bola splnená zákonná požiadavka alebo požiadavka na dodržiavanie predpisov alebo aby podporili kritický obchodný projekt. Príkladom Urgent Change je potreba aplikovať critical security patch (kritickú bezpečnostnú záplatu) na produkčný server na riešenie známej zraniteľnosti (vulnerability). Táto zmena by sa považovala za naliehavú, pretože je potrebné ju vykonať okamžite, aby sa predišlo potenciálnemu narušeniu bezpečnosti.

**6. Project Change (projektová)**

Ide o zmeny, ktoré sú súčasťou väčšieho projektu a sú riadené ako špecifický projektový typ. Môžu zahŕňať viacero tímov, zainteresovaných strán a môžu mať vyššiu úroveň zložitosti. Pred ich implementáciou si vyžadujú podrobné posúdenie, autorizáciu a schválenie. Napríklad implementácia nového systému riadenia vzťahov so zákazníkmi (Customer relationship management – CRM) pre spoločnosť. Táto zmena by sa považovala za Project Change, pretože je súčasťou väčšieho projektu a zahŕňa viacero tímov, zainteresovaných strán a má vyššiu úroveň zložitosti.

Change Management – CAB

Skôr než sa pohneme ďalej, trochu si to zrekapitulujeme.

**Change** (zmena) je akákoľvek úprava alebo zavedenie existujúcej alebo novej služby alebo jej časti. Nezáleží na tom, či sa premiestňuje jeden počítačový systém, premenúva sa databáza alebo či sa tisíce serverov presúvajú medzi silne chránenými podzemnými dátovými centrami – ak sa to mení, je potrebná change.

Zmeny môžu byť buď **normal** (bežné), alebo **emergency** (núdzové). Emergency changes sa vytvárajú ako reakcia na poruchu alebo hrozbu, zatiaľ čo Normal changes sú buď súčasťou každodennej činnosti, alebo nasledujú po nejakom strategickom rozhodnutí.

**Standard changes** sú založené na **Change Models** – štruktúrovaný prístup na riadenie konkrétnych typov zmien, ktorý zahŕňa vopred definované kroky, roly a postupy.

Standard changes síce vždy vychádzajú z Change modelu, ale **nie každá zmena** riadiaca sa Change modelom je zároveň Standard change. Môžu to byť aj Normal a dokonca Emergency changes.

Na pomoc žiadateľom v ich pokusoch požiadať o zmenu sa odporúča ponúknuť šablóny RfC, ktoré zlepšujú kvalitu údajov v žiadostiach o zmenu.

A nakoniec, zmeny musia byť schválené a správne naplánované, aby nedochádzalo ku konfliktom v plánovaní a aby ich vykonanie nenarušilo službu viac, ako je potrebné.

Ale pozor – toto je štvrtýkrát, čo sme spomenuli **approvals** (schválenia) bez vysvetlenia, ako schválenia fungujú!

Všeobecná zásada schvaľovania zmien je takáto: Zmenu musí posúdiť každý stakeholder (zainteresovaný subjekt) systému alebo komponentov alebo služby, ktoré sa menia, a len vtedy, ak **všetci súhlasia** s cieľom zmeny, plánom implementácie a harmonogramom, môže zmena pokračovať. Každý schvaľovateľ má teda právo veta.

Vzhľadom na množstvo zmien, ktoré môže poskytovateľ služby vykonať, ITIL definuje **Change Advisory Board – CAB** (Poradnú radu zmien) – skupinu ľudí, ktorí poskytujú rady, či má byť konkrétna zmena schválená. Typicky CAB diskutuje o viacerých zmenách za sebou – zvyčajne o zmenách, ktoré sú plánované na blízku budúcnosť.

Členovia CAB potrebujú na prijímanie rozhodnutí dostatok informácií, a ak pri Change takéto informácie chýbajú, štandardnou reakciou je odmietnutie zmeny; to zase motivuje Change requestors a Change managers, aby zhromaždili všetky informácie a aktualizovali **Change Record**, kým nebude uspokojivý.

O Change Management sme sa teda bavili už nejaký čas a ak ste to všetko vstrebali naraz, môže vám v hlave hučať ako v úli.

Kľúčové poznatky by mali byť tieto:

Po prvé, hoci zmeny sa dajú urobiť bez použitia procesu, iba proces môže zabezpečiť, že neprehliadnete nejaký kritický detail alebo že nedôjde ku konfliktom plánovania alebo chybám pri vykonávaní.

Po druhé, Change Management potrebuje bezchybné informácie. Ak sú rozhodnutia založené na zastaraných informáciách, budú zlé. To zase povedie k tomu, že infraštruktúra vôbec nefunguje alebo je otvorená pre všetky druhy zneužitia.

Po tretie, tento proces si vyžaduje úzku spoluprácu a otvorenú komunikáciu medzi technicky orientovanými ľuďmi a administratívne orientovanými ľuďmi. Technickí experti majú tendenciu podceňovať dôležitosť všetkej **papierovačky** vykonanej ich kolegami – ako je získanie všetkých tých schválení alebo vyplnenie všetkých podrobností v Change Records. Bez tohto papierovania a bez všetkých tých rozhodnutí by sa však infraštruktúra, ktorá sa mení, rýchlo zmenila na nezvládnuteľný chaos.

Týmto sa končí základná časť o procese Change Management. Tento proces budeme spomínať aj v budúcnosti, jednoducho preto, že mnohé iné procesy sú s ním silne prepojené a spoliehajú sa naň.

Service Asset and Configuration Management

Servise Transition obsahuje dva procesy, ktoré spravujú informácie. Pozrime sa na prvý z nich – Service Asset and Configuration Management.

**Service Asset and Configuration Management (SACM)**, niekedy v skratke nazývaný Config Management, má na starosti zhromažďovanie, údržbu, aktualizáciu a sprístupňovanie informácií o všetkých prvkoch infraštruktúry služieb a ich konfigurácii. Toto je opäť definícia, ktorú možno bude potrebné rozdeliť na časti, aby ste jej plne porozumeli:

**Service Assets** (Aktíva služieb) sú všetky komponenty, ktoré používa akákoľvek služba – vo svete IT by to mohli byť sieťové komponenty, ako sú routers a switches. Môžu to byť servery, úložné zariadenia, ale aj každá aplikácia, každý operačný systém a každá databáza. A ak sú zariadenia poskytnuté koncovým používateľom, ako sú mobilné telefóny alebo počítače, potom by to boli tiež aktíva.

Jednou životne dôležitou informáciou, ktorú treba pochopiť, je, že assets sa často kombinujú alebo zoskupujú, aby vytvorili iné aktíva; napríklad operačný systém, databáza a file system môžu byť skombinované na vytvorenie „server asset“ (serverového aktíva). Viaceré serverové aktíva môžu byť kombinované do klastra a tak ďalej a tak ďalej, až po celú infraštruktúru pre konkrétnu službu alebo zákazníka, ktorá je tiež aktívom. Je to výhodné, pretože nám to umožňuje odkazovať na mnohé prvky iba jedným identifikátorom.

Assets sú teda prvky, stavebné kamene služby. Ale hovorili sme o Change Managemente, takže vieme, že tieto assets sa neustále menia. Asset môže začať slúžiť jednému zákazníkovi špecifickým spôsobom, ale neskôr by mohla change zmeniť účel alebo dokonca spôsobiť, že bude slúžiť inému zákazníkovi. Len si predstavte, že zamestnanec odchádza z firmy a jeho počítač dostane nový zamestnanec – ide o zmenu konkrétneho majetku.

Je zrejmé, že ak chceme ako poskytovateľ služieb sledovať, ako naša služba funguje, ak chceme vedieť, ktorý prvok plní ktorú úlohu, musíme tiež uchovávať dve dôležité informácie:

Po prvé – ako je asset **nakonfigurovaný** – aké sú jeho nastavenia, kapacita a akékoľvek ďalšie konfiguračné charakteristiky, ako napríklad verzia operačného systému alebo jazykové nastavenia.

A po druhé – ako je to **spojené s inými** assetmi – fyzicky aj logicky (**relationship**)

Všetky tieto informácie spravuje služba Service Asset and Configuration Management. Tento proces uchováva záznamy o všetkých aktívach, ich konfiguráciách a ich prepojeniach s inými aktívami. Tieto informácie sprístupňuje aj iným procesom – napríklad Change Managementu.

SACM – CMDB

Hovorili sme o tom, že Change Records musia obsahovať bezchybné informácie, inak Change nebude schválená. SACM musí poskytovať Change Managementu informácie o assete a o tom, čo by mohol ovplyvniť, pretože to bude nevyhnutné na klasifikáciu ako napr. Minor alebo Major Change a tiež na plánovanie zmeny.

Predstavte si veľkú banku, ktorá potrebuje nainštalovať bezpečnostné záplaty na svoje servery. Ak by to urobili v pondelok o desiatej ráno, keď veľa zákazníkov chce mať prístup k svojim bankovým účtom, služba by bola nedostupná a zákazníci by boli veľmi nespokojní. Zmeny v bankovom sektore preto musia byť naplánované na nočný čas, zvyčajne medzi 2. a 5. hodinou ráno.

A nielenže Service Asset and Configuration Management poskytuje informácie iným procesom, ale tiež riadi aktualizáciu informácií – takže keď sa vykoná zmena ako súčasť procesu Change Management, spustí sa proces Config Management na aktualizáciu informácií o konkrétnom assete. Možno sa čudujete: Ale kde sú všetky tieto informácie uložené? Čo znamená „aktualizovať konfiguračné informácie“?

Všetky informácie sú uložené v **Configuration Management Databases** (databázach riadenia konfigurácie) alebo **C-M-D-B**. Tieto databázy sú súčasťou väčšieho systému s názvom **Configuration Management System**, ktorý okrem databáz pozostáva aj z nástrojov, ktoré pristupujú a manipulujú s dátami v databázach.

ITIL tiež definuje pojem **Configuration Item** (konfiguračná položka) alebo **C-I**, čo je jedinečný odkaz na reprezentáciu aktíva v CMDB. To zvyčajne vedie k tomu, že ľudia používajú výraz „CI“ (u nás céičko) namiesto „asset“. Záznam CI zvyčajne obsahuje aj jeho konfiguráciu a vzťahy s inými CIs. S vhodnými nástrojmi je možné, aby CMS zobrazoval logické mapy konfiguračných položiek a ich spojení – ale keďže ITIL nie je normatívny, nevyžaduje, aby to CMS robil.

Jedným zo spôsobov, ako sa pozrieť na Configuration Management Database, je pochopiť, že ide v podstate o inventár s množstvom pripojených detailov. Zatiaľ čo typický inventár v reštaurácii môže uvádzať len „20 stolov“ a „80 stoličiek“, ak by reštaurácia mala inventár v štýle CMDB, potom by mal každý stôl jedinečný názov – možno **stôl-nula-jeden**, až do **stôl dvadsať**. Každá stolička by mala tiež svoj vlastný identifikátor, napríklad **stolička-nula-jedna**, **stolička-nula-dva** atď. Okrem toho by tu bol aj asset s názvom **„sedenie-nula-jedna“**, ktoré by obsahovalo stôl-nula-jeden a štyri stoličky. A dokonca môže existovať **master asset** s názvom **„sekcia Sever“**, ktorý by obsahoval päť **„sedení“**. Aj keď to môže znieť prehnane, umožnilo by to zamestnancom rezervovať si **5 stolov a 20 stoličiek** v „severnej časti“ len niekoľkými kliknutiami ich rezervačného systému.

Majte na pamäti, že poskytovatelia IT služieb sa môžu potýkať s Major Changes, ktoré ovplyvňujú desiatky tisíc zariadení, a manuálne vyberať každé z nich by bolo časovo náročné.

SACM – PROCESY

SACM je interne rozdelený do troch podprocesov. Dva z nich sú orientované dlhodobejšie, zatiaľ čo tretí je určený na dennú prevádzku.

Prvý z procesov, **Configuration Identification** (Identifikácia konfigurácie) je proces, ktorý riadi Configuration Management System. Tento proces zabezpečuje, že náš CMS je schopný uchovávať a spracovávať všetky informácie o CI. Preto, keď je potrebné nové dátové pole, napríklad kvôli zmene technológie alebo právnych požiadaviek, proces identifikácie konfigurácie zabezpečí aktualizáciu CMS.

Druhý dlhodobý proces, **Configuration Verification and Audit** (Overenie a audit konfigurácie), zabezpečuje a vykonáva pravidelné audity, ktoré porovnávajú údaje v Configuration Management System so skutočnosťou a riešia prípadné nezrovnalosti. Je to veľmi podobné inventúre zásob, ktorú z času na čas robí každá organizácia, aby objavila stratené alebo poškodené vybavenie. Samozrejme, mať inteligentné zariadenia, ktoré môžu hlásiť svoj vlastný stav a konfiguráciu, je veľkou úsporou času.

Na dennej báze funguje proces **Configuration Control** (kontrola konfigurácie), ktorý riadi, kto a ako môže upravovať údaje v CMS. Configuration Control konkrétne zabezpečuje, že pridávanie CIs do CMDB alebo úprava záznamov existujúcich CIs je možná len po autorizácii a že nástroje CMS sledujú všetky tieto pridania a úpravy. Inými slovami, niekto, kto chce aktualizovať konfiguračné informácie, sa zvyčajne musí prihlásiť do nejakého systému a tento systém zaznamená, čo sa zmenilo a ako. To výrazne obmedzuje škody, ktoré by mohol zlomyseľný operátor spôsobiť jednej z najcennejších informačných zásob organizácie.

Knowledge Management

Druhý proces, ktorý spravuje informácie poskytovateľa služieb, sa nazýva **Knowledge Management**. Dalo by sa povedať, že kým Service Asset and Configuration Management je tu na správu informácií o komponentoch infraštruktúry, Knowledge Management sa stará o všetky ostatné znalosti. Poďme sa na to pozrieť bližšie.

**Knowledge Management** je príkladom procesu, ktorý bol do ITILu pridaný pomerne neskoro. Predtým, ako sa to stalo, bola téma zhromažďovania, uchovávania a zdieľania vedomostí roztrúsená v mnohých ďalších procesoch.

Skutočnosť, že znalosti sú teraz pod kontrolou jedného procesu, znamená dôležitosť znalostí a ich riadenia pre poskytovanie služieb.

**Knowledge Management zaisťuje, že znalosti a informácie v organizácii sú zhromažďované, uchovávané a poskytované ostatným.** Postupom času je tiež možné analyzovať informácie a získať ďalšie poznatky.

Knowledge Management sa snaží vyriešiť spoločný problém mnohých organizácií – problém znovuobjavovania vedomostí, alebo, ako hovoríme, „vymýšľať teplú vodu“.

Existujú dva hlavné dôvody, prečo sú vedomosti znovuobjavené:

Jedným z dôvodov je, že keď niekto objavuje vedomosť alebo informáciu, ostatní o tom nevedia, a keď je informácia objavená, nemusí byť okamžite zdieľaná so všetkými ostatnými. Toto je situácia, keď niekto strávi veľa hodín hľadaním najlepšieho analytického nástroja alebo učením sa, ako naprogramovať makro v tabuľkovom procesore, a potom sa dozvie o inom tíme, ktorý to riešil pred rokmi.

Druhým častým dôvodom sú odborníci, ktorí opúšťajú organizáciu a odnášajú si svoje znalosti so sebou. To priamo súvisí s prvým dôvodom, ale zaslúži si osobitnú zmienku - ak by sa experti podelili o všetky svoje poznatky s ostatnými, potom by ich odchod nebol až takou ranou pre zachovanie vedomostí. Časť toho, čo robí odborníkov tak dôležitými, je množstvo vedomostí v ich mysli a pamäti, čo vytvára nechuť všetko zdieľať alebo zapisovať.

Knowledge Management - SKMS

Kowledge Management nie je rozdelený na podprocesy a definuje pojem **Service Knowledge Management System** alebo **S-K-M-S.**

SKMS je zvyčajne súbor nástrojov, ktoré umožňujú ukladanie, manipuláciu, analýzu a získavanie všetkých druhov informácií a znalostí. Je bežné, že každý z týchto nástrojov je špecializovaný na určitý typ znalostí – napríklad jeden nástroj, ktorý spravuje pracovné pokyny a návody, ďalší nástroj, ktorý môže podporovať pracovné postupy, a ďalší nástroj na sledovanie projektov a ich aktuálnych stavov.

Na ilustráciu témy sa ešte raz pozrime na našu reštauráciu. Aké druhy vedomostí by teda reštaurácia chcela spravovať?

Je tu zrejmá ponuka jedál, ktoré ponúkajú. Ku každému z jedál, ktoré ponúkajú, je priložený recept na jeho prípravu a aj tieto recepty by sa mali riadiť.

Podobné pravidlá platia pre zoznam nápojov, ktoré ponúkajú, a tie nápoje, ktoré si vyžadujú prípravu, vrátane faktov ako „ako dlho pred podávaním otvoriť fľašu vína, aby mohla dýchať“, môže každý potrebovať samostatný záznam. Samozrejme, využívanie moderných technológií, ako je pridávanie teplomerov do fliaš, by mohlo podporiť riadenie znalostí.

Ale to určite nie sú všetky znalosti, ktoré by mala reštaurácia ovládať.

Mali by napríklad existovať podrobné pokyny pre všetkých rôznych zamestnancov, ako majú konať. Tie by mali poskytovať usmernenia pre bežné aj výnimočné situácie – a tam, kde je to vhodné, aj pre prístup k rôznym typom zákazníkov. Zákazníci s malými deťmi sa budú na prvom rande správať inak ako pár a päť dobre oblečených manažérov s viditeľne odlišným kultúrnym zázemím bude potrebovať iný prístup ako skupina štamgastov, ktorí si vždy objednávajú rovnakých pár jedál.

Ďalším typickým príkladom vedomostí, ktoré treba spravovať, je zoznam kontaktov na VIP zákazníkov. Pamätajte - ITIL neprikazuje, takže je úplne možné a v skutočnosti veľmi praktické mať tento zoznam nahraný v telefóne reštaurácie. Týmto spôsobom, ak zavolá VIP zákazník, zobrazí sa jeho meno a môže byť privítaný spôsobom, ktorý očakáva a vďaka ktorému sa bude cítiť dôležito.

Podobne je rozumné mať zoznam kontaktov na záchrannú službu a bezpečnostnú agentúru pre prípad problémov – opäť sú to znalosti a treba ich riadiť.

**Knowledge Management** môže často nahrádzať iné procesy. Pamätáte si na Supplier Management zo Service Strategy? Pre začínajúcu spoločnosť, ktorá potrebuje mať funkčnú tlačiareň s dostatkom atramentu alebo laserového tonera, môže byť prehnané mať proces Supplier Managment, predkladať verejné žiadosti o ponuky a pravidelne kontrolovať dodávateľský reťazec; aspoň na začiatku si pravdepodobne zaobstarajú tlačiareň, možno si na internete vyhľadajú najlepšiu cenu atramentových či tonerových kaziet a zistené informácie si nechajú po ruke nabudúce.

Na záver kapitoly o manažmente znalostí sa hovorí: Ľudia začnú zálohovať, keď stratia dáta po druhýkrát.

Organizácia si zvyčajne nie je vedomá všetkých vedomostí, ktoré má, až kým časť z nich nezmizne. A keď sa tak stane, zodpovední si vo všeobecnosti uvedomia, ako mali byť poznatky zachované. Ak sa tieto opatrenia zavedú, Knowledge Management sa zlepší.

Release and Deployment Management

Teraz, keď vieme, ako funguje Change Management a kde si organizácia uchováva svoje informácie a znalosti, môžeme prejsť k zložitejším oblastiam Service Transition.

Majte na pamäti, že ide o pokročilé témy a vo všeobecnosti ich riadia skúsení manažéri procesov. Neočakáva sa od vás, že si zapamätáte celú terminológiu a budete ju vedieť bez váhania citovať.

Kvôli zložitosti tiež nie je jednoduché uvádzať príklady analógie zo sveta mimo IT.

R&DM – základné pojmy

**Release and Deployment Management** je komplexný proces, ktorý sa používa, keď je v prostredí potrebné vykonať väčší počet zmien. Typickým príkladom je migrácia infraštruktúry zákazníka medzi lokalitami alebo od iného poskytovateľa služieb k nám. V takom prípade by bolo presúvanie každej položky konfigurácie oddelene, v samostatnej Change, nepraktické a nemali by sme jasný prehľad o celkovej situácii.

Najdôležitejšie je, že Release Management sa snaží **minimalizovať prerušenie služby spôsobené rozsiahlymi zmenami infraštruktúry**. Myslite na výstavbu nového nákupného centra uprostred mesta – cesty budú musieť byť uzavreté, zem vykopaná, presunuté vodovodné potrubia a podobne. Ak by stavebná firma jedného dňa začala kopať bez akejkoľvek prípravy, občanom by narušenie života veľmi vadilo. Preto je potrebné vopred zdieľať informácie, označiť a oznámiť alternatívne trasy, zabezpečiť núdzové zásobovanie vodou atď. Všetky tieto aktivity by potom spravoval Release Management.

**Releases** sa skladajú z **Release Units** (jednotiek vydania) - jednej alebo viacerých. Každá Release Unit obsahuje viacero **CIs** (konfiguračných položiek), ktoré je potrebné presunúť, zvyčajne s nejakým zastrešujúcim spoločným účelom. Takže celý Release môže obsahovať napríklad 3 Release Units – jednu pre databázové systémy, jednu pre úložné systémy a jednu pre aplikačné systémy.

V príklade s výstavbou by jedna Release Unit riadila samotnú výstavbu, ďalšia jednotka by sa zamerala na presmerovanie dopravy a ďalšie jednotky by riešili zásobovanie vodou, tlmenie hluku a možno aj komunikáciu smerom k verejnosti.

V rámci Release Unit môže existovať **Development Work Order** (objednávka vývojárskej práce) – to je spúšťač pre **Application Development** (vývoj aplikácií) na prispôsobenie alebo vývoj systému, ako je napríklad aplikácia. Alternatívne môže Release Unit obsahovať **Installation Work Order** (objednávka inštalácie), ktorou sa objednáva inštalácia konfiguračných položiek.

Je dôležité pochopiť, že Releases sa primárne zameriavajú na tie zmeny, ktoré musia byť najprv postavené vo vývojovom prostredí – **development environment**, potom testované v testovacom prostredí – **testing environment** a až potom môžu byť nasadené - **deployed**, čo znamená - presunuté do živého prostredia – **live environment**.

R&DM – fázy

Release Management zvyčajne začína plánovaním vydania – **Release Planning**. V tomto kroku sa všetky potrebné zmeny rozdelia a priradia k správnym Release Units a definuje sa ich oblasť. Potom je možné tieto Release Units správne naplánovať – kedy budú zostavené, kedy budú testované a nasadené.

Potom sa všetky Development Work Orders a Installation Work Orders predložia počas fázy **Release Build** (zostavovania vydania). V tejto fáze sa nakupuje hardvér a softvér, robí sa vývoj, vykonáva sa prispôsobenie - customization. Na konci by mali byť všetky komponenty prítomné vo vývojovom prostredí a pripravené na testovaciu fázu – **testing phase. Testovacia fáza** prebieha s cieľom overiť, či nový release funguje podľa očakávaní a neohrozí stabilitu IT prostredia. Zahŕňa rôzne typy testovania, ako napríklad **funkčné, výkonové, bezpečnostné a používateľské testy**, ktoré zabezpečujú hladké nasadenie služby.

**Release Deployment** je potom ďalšou fázou, v ktorej sa CIs (konfiguračné položky) presunú z testovacieho prostredia do živého prostredia. Okrem toho sú používatelia vyškolení v používaní alebo prevádzke komponentov a podľa potreby sa poskytuje dokumentácia.

Napriek maximálnemu úsiliu môžu pri nasadzovaní vydania nastať komplikácie, a preto je k dispozícii podpora **Early Life Support** na vyriešenie akýchkoľvek problémov počas prvého obdobia ostrej prevádzky. Je to dočasná fáza podpory po zavedení novej služby alebo veľkej zmeny. Počas tohto obdobia sa služba **pozorne monitoruje** a prípadné problémy sa **rýchlo riešia**, aby všetko fungovalo hladko. Zároveň pomáha operačnému tímu zoznámiť sa s novou službou, kým prejde do bežnej podpory. Taktiež, ak bol poskytovateľ služby informovaný o veciach, ako je chýbajúca dokumentácia alebo nejasné školenie, aj tieto nedostatky sa riešia v tejto fáze.

Nakoniec **Release Closure** (Ukončenie vydania) je posledný krok v release procese, kde sa všetko uzatvára po úspešnom zavedení novej služby alebo zmeny. Zahŕňa **kontrolu, či release splnil svoje ciele**, zdokumentovanie získaných poznatkov a zabezpečenie, že všetky zostávajúce problémy sú odovzdané do bežnej podpory. Keď je potvrdené, že všetko funguje stabilne, release sa oficiálne uzavrie.

V tejto fáze je nutné sa presvedčiť, že Configuration Database (konfiguračná databáza) obsahuje aktuálne informácie o všetkých CIs (konfiguračných položkách).

Application Development and Customization

**Application** **Development** (Vývoj aplikácií), niekedy tiež označovaný celým názvom Application Development and Customization (Vývoj a prispôsobenie aplikácií), je časťou Service Transition, ktorá sa zameriava na vývoj a úpravy softvéru. **Často sa spúšťa z Release Managementu.**

**Software Development** (Vývoj softvéru) je komplexná téma a existuje množstvo návodov, princípov a metodík, ktoré sa snažia, aby bol takýto vývoj efektívny, efektívny a zvládnuteľný.

Z tohto dôvodu knihy ITIL neobsahujú veľa informácií pre Application Development a experti v tejto oblasti musia hľadať informácie inde.

**Application** **Development** je mimo obsahu ITIL Foundation, takže aby ste mali len rýchly prehľad, tu je všeobecný priebeh tohto procesu.

Vo väčšine prípadov sa identifikuje **potreba** aplikácie, niekedy nazývaná aj **trhová potreba**. Môže to byť „Ľudia by chceli posielať e-maily zo svojich mobilných telefónov“, „Ľudia by si chceli objednať pizzu cez SMS“ alebo „Hráči chcú zabiť milióny zombíkov v plne trojrozmernej hre“.

Po identifikácii potreby vývojári prejdú do **planning phase** (plánovacej fázy), v ktorej rozdelia svoju aplikáciu určenú na **uspokojenie potreby** na zvládnuteľné časti a potom sa na základe svojho know-how a odhadov rozhodnú o zdrojoch potrebných na tieto kúsky.

Pre každý z týchto kúskov alebo prvkov aplikácie sa musí najskôr urobiť **design** (návrh), po ktorom nasleduje opakovaná sekvencia implementácie, testovania a dokumentácie – **implementing, testing, and documenting.**

Jedna múdrosť, ktorú sa možno naučiť z vývoja aplikácií a použiť ju inde, je koncept vývojového dlhu – **Development Debt**. Development Debt je výraz pre všetky skratky, ktoré boli použité počas vývoja a nikdy sa nezmenili na ich úplné cesty – povestné rezanie zákrut.

Napríklad jedna z prvých vecí, ktorá sa pod časovým tlakom vynecháva, je **dokumentácia** funkčnosti implementovaného kódu alebo funkcií – vývojár jednoducho šetrí čas tým, že ho netrávi písaním podrobnej dokumentácie.

Múdrosť alebo poučenie znie, že **development debt** časom kumuluje **úroky** a že v určitom momente ich niekto musí splatiť. Vývojár teda môže ušetriť 2 týždne času tým, že nebude písať dokumentáciu, ale o rok neskôr, keď zákazníci požadujú nové funkcie, ten istý vývojár môže potrebovať ďalších 10 týždňov, aby zistil, ako funguje jeho pôvodný kód.

To platí aj pre iné procesy – na konci incidentu môžete ušetriť pár minút tým, že svoje riešenie podrobne nepopíšete a záznam jednoducho zatvoríte. Ale o mesiac neskôr, keď budete čeliť rovnakému incidentu, nebudete môcť vyhľadať pôvodné riešenie a len ho zopakovať; budete musieť znovu vymýšľať teplú vodu, znovu objaviť toto riešenie, čo bude pravdepodobne trvať oveľa dlhšie.

Aby sme mohli pokračovať v procese vývoja, po implementácii, testovaní a zdokumentovaní všetkých prvkov aplikácie je možné aplikáciu nasadiť - **deploy**, čo je oficiálny názov pre jej spustenie alebo sprístupnenie zákazníkom a používateľom.

A pokiaľ je aplikácia dostupná, treba ju udržiavať – **maintain**, opravovaním chýb, zlepšovaním používateľského zážitku a zvyčajne aj vylepšovaním funkčnosti.

Predtým, ako prejdeme na nový proces, len rýchla rekapitulácia na upevnenie pochopenia: Release and Deployment Management sa zameriava na balíky zmien nazývané Releases, ktoré je potrebné vykonať koordinovaným spôsobom. Tieto Releases sú rozdelené do jednej alebo viacerých logicky prepojených Release Units a ako súčasť týchto jednotiek môže byť spustený Application Developmnent and Customization, aby sa vyvinul alebo upravil nejaký počítačový systém pre naše potreby. Releases prechádzajú fázou budovania, fázou testovania a nakoniec fázou nasadenia **(building, testing, deployment)** – a Releases sa nemôžu presunúť do ďalšej fázy, kým nie sú úspešne dokončené všetky činnosti pre aktuálnu fázu. Po úspešnom nasadení vydania Early Life Support opraví všetky nedostatky, ktoré sa objavia, a podľa potreby upraví školenia alebo dokumentáciu.

Service Validation and Testing

Mali sme testovanie v rámci Release and Deployment Management:

Každý release má svoje vlastné testovacie fázy (napr. unit testing, system testing, user acceptance testing).

Toto testovanie sa vykonáva pred nasadením do produkcie, aby sa overilo, že release funguje správne.

Teraz môžeme obrátiť našu pozornosť na zábavnejší proces s názvom **Service Validation And Testing** (Overenie a testovanie služby).

Toto je samostatný proces, ktorý sa zameriava na **validáciu a testovanie služby ako celku**. Cieľom SV&T je zabezpečiť, že nová alebo zmenená služba je plne pripravená na produkčné nasadenie a poskytovanie očakávanej hodnoty zákazníkom.

Môže prebiehať **počas a aj po** Release and Deployment Management, ale často sa uskutočňuje **pred nasadením** služby do produkcie, aby sa overilo, že splnila všetky požiadavky. Release Management vykonáva svoje vlastné testovanie, ale po Release closure môžu skryté chyby spôsobiť, že služba bude neskôr fungovať zle alebo s chybami.

Po ukončení release (**Release Closure**) sa môžu vykonávať ďalšie validácie a testy, napr. early life support (ELS), ktoré sledujú výkon novej služby v reálnom prostredí. Ak sa vyskytnú problémy, môžu byť potrebné dodatočné zmeny alebo rollback.

Service Validation and Testing je **oddelený** od Release and Deployment Management, aby sa zabezpečila objektívna kontrola kvality a minimalizovali riziká spojené s nasadením novej alebo zmenenej služby.

Možno ste videli tento konkrétny IT vtip:

Softvérový tester vstúpi do baru. Objedná si pivo. Objedná si **mínus** jedno pivo. Objednáva **nula** pív. Objednáva **deväť-deväť-deväť-deväť-deväť-deväť** pív. Objedná si **jaštericu**. Snaží sa odísť bez zaplatenia.

Tento vtip ukazuje len zlomok všetkých možných vstupov, ktoré môže služba dostať – platné aj neplatné vstupy. Je to užitočná analógia toho, čo testeri robia, aby odhalili chyby v službe. Chyby v službe sa môžu pohybovať od drobných, ako je napríklad služba, ktorá vám umožňuje objednať si „nulu“ niečoho, až po veľké chyby, napríklad vám umožňujú objednať mínus päťsto mobilných telefónov a potom vám ponúknu, že vám pošlú peniaze späť, pretože ste zdanlivo preplatili o stotisíc dolárov.

Nedostatky v službách môžu viesť k odhaleniu osobných údajov neoprávneným používateľom alebo dokonca k ich hromadnému odcudzeniu a následnému predaju na čiernom trhu.

Preto je dôležité overiť, či je služba poskytovaná iba oprávneným používateľom a či používateľ dostáva presne to, čo je oprávnený prijímať. **Service Validation And Testing** služby je proces, ktorý sa zameriava práve na toto overenie.

Naivným prístupom k testovaniu služieb je nechať testovacích používateľov používať službu ľubovoľným spôsobom a požiadať ich, aby nahlásili čokoľvek neobvyklé - alebo aby zaznamenali svoje akcie a potom to vyhodnotili inak.

Tento prístup k testovaniu má viacero problémov:

Po prvé, neexistuje žiadna záruka, že používatelia otestujú všetky platné interakcie so službou – možno preto, že niektoré platné použitie je také nezvyčajné alebo zriedkavé, že žiadny z testovacích používateľov by toto použitie sám neskúšal. Ak napríklad elektronický e-shop dáva testovacím používateľom možnosť objednať si akúkoľvek položku zo skladu, atraktívne typy mobilných telefónov budú fiktívne objednané mnohokrát, zatiaľ čo niektoré špecializované produkty, ako napríklad outdoorový mobil alebo mobil pre nevidomých používateľov, by sa vôbec neobjednali.

Po druhé, testovací používatelia sa budú vo všeobecnosti správať ako čestní zákazníci a nebudú sa snažiť úmyselne narušiť službu alebo sa dostať na miesta, kam nemajú chodiť.

Napríklad polia webového formulára môžu mať obmedzenú kapacitu pre text, ktorý je možné do nich zadávať, a bez pokročilých nástrojov bežný používateľ nedokáže zadať dlhší text. Zadávanie dlhšieho textu do poľa webového formulára je však jedným z najbežnejších vektorov útoku pre jednotlivcov so zlými úmyslami.

A napokon, tento prístup nie je možné správne riadiť z hľadiska nákladov a trvania – nie je možné určiť, ako dlho to bude trvať a či bol test dostatočne dôkladný.

Service Validation and Testing – Test Models

Aby bolo testovanie služby zvládnuteľné, ITIL používa **Test Models** (testovacie modely), ktoré sú podrobným popisom toho, ako bude release – alebo služba – testovaný. Presnejšie povedané, Test Model musí špecifikovať prístup, ktorý sa použije na testovanie.

**Test Model Definition** (Definícia testovacieho modelu) je procesný krok, ktorý definuje model; samotný Test Model sa **vytvára** počas Release planning. Ako môžete vidieť, Service Validation and Testing beží – alebo sa spúšťa – súbežne s Release Managementom.

Keď je test definovaný a vytvorený, ďalším krokom je **Release Component Acquisition** (získanie komponentu vydania). Tento krok sprístupní komponenty, ktoré musia byť testované. Pred skutočným testom sa vykoná počiatočné posúdenie, aby sa overilo, či komponent nie je žiadnym spôsobom chybný (toto je obzvlášť dôležité pre hardvérové ​​komponenty, ako sú senzory, pretože chybný senzor môže ľahko prejsť niektorou časťou skutočného testu tým, že nebude mať žiadne **falošne pozitívne** výsledky).

Po akvizícii nasleduje **Release Test** - skutočné testovanie podľa testovacieho modelu. Platné aj neplatné vstupy sú testované a schopnosť komponentu akceptovať všetky platné vstupy pri odmietnutí všetkých neplatných vstupov je prísne sledovaná a dokumentovaná.

Ak sa služba poskytuje konkrétnemu zákazníkovi, zástupca tohto zákazníka je zvyčajne prítomný počas testovania a potvrdzuje, že komponenty testom prešli. Ak je služba zacielená na veľa zákazníkov – napríklad služba mobilného telefónu alebo kaviareň, ktorá predáva kávu tisíckam ľudí, potom je možné vybrať testovaciu skupinu a použiť ju na zastúpenie všetkých zákazníkov.

Po dokončení všetkých Release Tests – nezabudnite, že tento krok kontroluje **jednotlivé** komponenty – sa otestuje **celá služba ako celok**. Toto sa nazýva **Service Acceptance Testing** (akceptačné testovanie služby) a je zamerané na odhalenie akýchkoľvek chýb, ktoré sú spôsobené interakciami medzi komponentmi.

**Napríklad**: nabíjačka pre mobilný telefón môže fungovať perfektne, headset pre mobilný telefón môže fungovať perfektne, telefón môže perfektne telefonovať, ale môže sa stať, že ak telefón pripojíte k nabíjačke a pripojíte headset a prijmete hovor, batéria sa preťaží a telefón sa vznieti. Áno, to sa naozaj stalo.

Podobne, jednotlivé komponenty ako je webová aplikácia, databáza a úložný systém, môžu všetky fungovať uspokojivo, ale keď sa do webovej aplikácie prihlási tisíc používateľov súčasne, celý systém sa spomalí a prestane reagovať.

Ak **Service Acceptance Testing** neodhalí žiadne nedostatky, potom najdôležitejším krokom je, že zástupca zákazníka podpíše záväzné akceptovanie služby – inými slovami, zákazník potvrdí, že služba spĺňa jeho očakávania a že ju akceptuje. Zákazníkovi to ukladá zmluvný záväzok používať službu a platiť za ňu.

Project Management – Transition Planning and Support

Nad všetkými týmito procesmi v Service Transition vyčnieva **Project Management** (projektový manažment), tiež známy ako **Transition Planning and Support** (plánovanie a podpora tranzície). Bude to posledný proces, ktorý podrobne popíšeme v tejto kapitole.

Alternatívny názov môže byť aj Transition And Transformation.

Keďže ITIL je zbierkou osvedčených postupov, nehovorí príliš podrobne o tom, ako riadiť projekty – používa sa niekoľko osvedčených metodík, ako napríklad *PMBOK* (Project Management Body of Knowledge - je súbor štandardnej terminológie a smerníc (súbor znalostí) pre projektový manažment) alebo *PRINCE2* (PRojects IN Controlled Environment - je metóda štruktúrovaného projektového manažmentu a program certifikácie odborníkov. PRINCE2 kladie dôraz na rozdelenie projektov do manažovateľných a kontrolovateľných etáp).

Z tohto dôvodu považujte týchto pár minút len za malý úvod do sveta projektového manažmentu.

**Transition Planning and Support** v ITILe má za cieľ plánovať a koordinovať zdroje pre Releases, najmä tie veľké, účinným a úsporným spôsobom. Všeobecne povedané, tento proces sleduje čas, kvalitu a náklady na Major Releases a zabezpečuje, aby tieto releases neprekročili svoje limity – inými slovami, aby neprekročili rozpočet, netrvali dlhšie, ako sa plánovalo a dodali očakávanú kvalitu.

Na tento účel sú k dispozícii 4 čiastkové rocesy.

**Project Initiation** (začatie projektu) je zvyčajne otvárací čiastkový proces, v ktorom sa identifikujú **stakeholders** (zainteresované strany), nominuje sa **Project Team** (projektový tím) a **responsibilities** (zodpovednosti), vyčlenia sa **resources** (zdroje) a vykoná sa **risk analysis** (analýza rizík), len aby sme vymenovali niektoré z vykonaných činností.

Ihneď potom začína **Project Planning and Coordination** (plánovanie a koordinácia projektov) ako prebiehajúca aktivita. To pozostáva z plánovania všetkých činností a ich koordinácie podľa smerníc organizácie pre Project Management.

**Project Control** (Kontrola projektu) je špeciálnym krokom, ktorý pravidelne kontroluje priebeh projektu, ako aj jeho zdroje, použité aj zostávajúce, a môže iniciovať nápravné opatrenia, ak projekt potrebuje takúto korekciu.

A keďže žiadny projekt nemôže fungovať bez reportingu a komunikácie – **reporting and communication**, ITIL definuje aj špeciálny podproces pre tieto potrebné činnosti.

Celkový obraz

Aby ste lepšie porozumeli, pozrime sa, ako tieto procesy do seba zapadajú.

**Transition Planning and Support** je najvyšší proces – Transition manager môže byť priamo podriadený predstavenstvu poskytovateľa služieb, keďže takýto Transition Manager má na starosti celú tranzíciu (prechod) – rozsiahlu zmenu infraštruktúry.

Keď sa tranzícia plánuje, jej aktivity sa stanú vydaniami **- Releases**, ktoré spravuje **Release And Deployment Management**. Každé z týchto vydaní môže spravovať špecializovaný Release Manager, ktorý potom podáva správy Transition Mangerovi. Release Management je preto o úroveň nižšie ako Transition planning and Suppport.

Ako releases pokračujú, **Changes** môžu byť vyžiadané alebo odoslané do Change Managementu. Alternatívne môže byť potrebný Application Development and Customization na prispôsobenie komponentov našim potrebám. Tieto procesy sú preto o jednu úroveň nižšie ako Release Management.

Service Validation and Testing sa použije na kontrolu správnosti Releases; Transition Manager bude pravdepodobne jednou zo zainteresovaných strán informovaných o prebiehajúcich validáciách a môže byť dokonca požiadaný, aby potvrdil, že validácia bola úspešná. Ak overenie služby zistí nedostatky, Transition Manager môže rokovať so zástupcami zákazníka, ako postupovať.

Samozrejme, všetky zmeny v infraštruktúre musia byť zdokumentované v Configuration Management Database, ktorá je v kompetencii Service Asset And Configuration Managementu.

Knowledge Management neustále podporuje všetky tieto procesy so všetkými znalosťami, ktoré organizácia zhromaždila.

Transitions sú obzvlášť silným zdrojom vedomostí alebo možno aj **múdrosti,** pretože predstavujú najkomplexnejšie súbory činností vykonávaných v záujme obsluhy zákazníkov, a preto sú najštedrejším zdrojom poučení.

Zhrnutie

Týmto sa končí náhľad do Service Transition. Hlavné závery z toho sú:

1. Kedykoľvek sa musí zmeniť niektorá časť našej služby, táto časť vstúpi do Service Transition a v určitom bode sa vykoná **Change.**
2. So zvyšujúcou sa komplexnosťou potrebných zmien sa zapájajú **ďalšie procesy**, ktoré **poskytujú podporu** z hľadiska kontroly kvality, dostupnosti zdrojov a rozpočtovania. **Release Management** používame na minimalizáciu prerušenia služby počas týchto väčších zmien. Pre skutočne rozsiahle zmeny, ako je úplné premiestnenie infraštruktúry medzi geograficky oddelenými oblasťami, je **Transtition Planning and Support** najsilnejším z týchto procesov.
3. Aby sme mohli robiť správne rozhodnutia, potrebujeme dokonalé informácie. **Configuration Database** (Konfiguračná) databáza obsahuje informácie o prvkoch infraštruktúry služieb, ich konfigurácii a ich vzťahoch, zatiaľ čo **Service Knowledge Management Systems** riadia všetky ostatné znalosti, ako sú pracovné postupy, procesy, zoznamy kontaktov, pracovné toky, získané skúsenosti atď.