

Obsah

Motivace	9
Úvod	13

ČÁST I.: Definice pojmů, současný stav provozu, údržby a podpory IS

17

1.

Vymezení pojmů a problému, současný stav	19
1.1 IT systém, informační systém, IT služba	19
1.1.1 Informační systém versus IT služba	20
1.1.2 Pojetí IT služby	21
1.1.3 Katalog IT služeb	24
1.1.4 Vytvoření katalogu IT služeb	25
1.1.5 Poskytování softwaru jako služby (SaaS)	26
1.2 Problémy provozu, podpory a údržby IT systémů	27
1.3 Rozsah podpory a údržby IT systémů	30
1.4 Metodiky, postupy, standardy	35
1.4.1 Správa IT služeb (ITSM)	39
1.4.2 Standard ISO 20000	50
1.4.3 Standard IEEE 1219	51
1.4.4 Standard ISO 12207	52
1.4.5 Microsoft Operations Framework (MOF)	54
1.4.6 Metodika MANTEMA	55
1.4.7 Enterprise Unified Process	56
1.4.8 CobiT	59
1.5 Shrnutí tradičních přístupů a jejich diskuse	60
1.6 Souhrn obecných problémů údržby IT systémů	63

ČÁST II.: Agilní a štíhlý provoz a údržba IT systémů

65

2.

Agilní a štíhlý provoz a údržba	67
2.1 Agilní vývoj a údržba IT systémů	68
2.2 Principy agilního a štíhlého provozu a údržby	72
2.2.1 Princip (1): Více disciplíny, méně byrokracie	74
2.2.2 Princip (2): Spolupráce a komunikace, systémový pohled	79
2.2.3 Princip (3): Proaktivita a učení	87
2.2.4 Princip (4): Riziky řízený přístup	91
2.3 Knihovna praktik	99
2.3.1 Iterativní přístup	101

2.3.2	Podnikové scénáře	103
2.3.3	Rotace	104
2.3.4	Boj s mnohoznačností	106
2.3.5	Defenzivní programování	107
2.3.6	Konvence pro zápis kódu	108
2.3.7	Agilní dokumentace	109
2.3.8	Testy řízený přístup a vývojářské testy	112
2.3.9	Refaktoring zdrojového kódu	118
2.3.10	Párová práce (Pair working)	120
2.3.11	Učení se na reálné práci	123
2.3.12	Neustálá integrace (<i>Continuous Integration</i>)	124
2.3.13	Retrospektiva	126
2.3.14	Denní schůzky napříč týmy (<i>Daily meetings</i>)	128
2.3.15	Odhadování, hlasování (<i>Planning poker</i>)	129
2.3.16	Vizualizace	131
2.4	Knihovna technik	135
2.4.1	Změny v kódu trvají věčnost	135
2.4.2	Techniky pro odstraňování závislostí v kódu	137

3.

Anti-vzory	141
3.1 Katalog anti-vzorů	141
3.2 Vývojářské anti-vzory	143
3.3 Anti-vzory architektury	144
3.4 Anti-vzory řízení	146
3.5 Procesní anti-vzory	147

ČÁST III.: Měkké aspekty provozu a údržby 149

4.

Měkké aspekty v provozu a údržbě	151
4.1 Typologie osobnosti MBTI	152
4.1.1 Zaměření: introvert vs. extrovert	153
4.1.2 Vnímání: intuice vs. smysly	154
4.1.3 Rozhodování: myšlení vs. cítění	155
4.1.4 Orientace funkcí: vnímání vs. usuzování	156
4.1.5 Temperamenty	156
4.1.6 Význam a přínos pro tým provozu a údržby	157
4.1.7 Test a další zdroje	159
4.2 Základy kognitivních věd	159
4.3 Tým	164
4.4 Motivace	167
4.4.1 Motivace a uspokojení z práce v provozu a údržbě	169
4.4.2 Odměňování a motivace	172

4.5 Koučing a mentoring	174
4.5.1 Talent, motivace a růst	177
4.6 Vedení a vůdcovství (Leadership)	179
4.7 Oběd: sdílení a sonda do stavu týmu	184

ČÁST IV.: Prostředí a celkový obraz

5.	Kontrakty	187
	5.1 Iterativní model kontraktu	191
	5.2 Jak tyto principy uplatnit v praxi?	193
	5.3 Shrnutí problematiky agilních kontraktů	197
6.	Governance – řízení podniku a IT	199
	6.1 Dvě ukázky neexistence governance	201
	6.2 Liché iniciativy managementu pro zlepšení kvality	204
	6.2.1 Důležitost zpětné vazby	206
	6.3 Prozřetelný pan Brooks	207

ČÁST V.: Praktická implementace agilního a štíhlého provozu a údržby

7.	Jak na implementaci agilního a štíhlého provozu, údržby a podpory v praxi?	211
	7.1 Postup implementace	213
	7.1.1 Krok 1: Analýza a sběr zkušeností, ponaučení	214
	7.1.2 Krok 2: Výběr relevantních agilních a štíhlých praktik	220
	7.1.3 Krok 3: Definice postupu	222
	7.1.4 Krok 4: Denní podpora implementace	225
	7.1.5 Krok 5: Analýza a sběr zkušeností	225
	7.2 Náklady a rizika daného přístupu	226
	7.3 Příklad implementace postupu – případová studie	228
	7.4 Příklad převzetí IT služby – případová studie	232
	7.5 Ověření přístupu (verifikace a validace)	236
	7.5.1 IT služba 1	236
	7.5.2 IT služba 2	238
	7.5.3 IT služba 3	240
	7.5.4 Dotazníkové šetření	243
	7.5.5 Neúspěšné implementace přístupu	244

ČÁST VI.: Podpora nástroji a budoucí vývoj 247

8.

Formální expertní systém 249

8.1 Metoda rozhodování 252

8.2 Expertní systém 254

8.3 Příklad aplikace 254

8.4 Další vývoj expertního systému 259

9.

Budoucí vývoj v oblasti provozu a údržby 261

9.1 Systémy založené na pravidlech 261

9.1.1 Pojmy a architektura pravidlových systémů 262

9.1.2 Způsob vývoje aplikace 264

9.1.3 Implementace změny 265

9.2 IBM přístup: Brownfield vývoj 265

9.3 Cloud computing 269

9.4 Shrnutí moderních přístupů 272

Závěr 273

Literatura 274

Seznam obrázků 281

Seznam tabulek 285

Příloha 286

Rejstřík 287

Motivace

Každý, kdo pracuje ve vývoji, provozu, podpoře či údržbě IT systémů (ať již komplexních informačních systémů, základního softwaru, IT infrastruktury či outsourcovaných aplikačních IT služeb) je denně konfrontován s incidenty a jinou operativní činností, díky nimž nemá čas řešit věci systémově, tj. hledat příčiny problémů, automatizovat a vylepšovat poskytované služby. Říkáme, že **pracuje v reaktivním módu**. Důsledkem toho klesá motivace pracovníků, ti schopní odcházejí, mnohdy nejsou peníze na školení či nákup nových softwarových licencí, a pokud je někdo poslán na školení, neví, jak nově nabyté poznatky zavést v jeho konkrétním prostředí, kde začít s realizací změny.

Další zkušeností je jistá **odpojenost manažerů** na různých úrovních **od produkčních týmů**. Tato odpojenost je projevuje několika způsoby:

- manažeři se na problémy neptají, chtějí slyšet, že je všechno v pořádku a že dodáme na čas, za dohodnutou cenu a v očekávané kvalitě;
- manažeři problémy nevidí, i když na ně členové týmu upozorňují, nebo je ignorují ať již z důvodu nadřazenosti, zbožného přání, že všechno nějak dopadne, či z důvodu neznalosti disciplín vývoje, provozu a údržby IT systémů.

Jednou z příčin této odtrženosti je však také neschopnost členů produkčních týmů problémy vizualizovat v „manažerské“ řeči, tj. co vše je pro realizaci třeba udělat, jaký bude dopad na finance (náklady a přínosy), datum dodání či další sledované ukazatele.

V neposlední řadě podle naší zkušenosti **rigorózní formální procesy brání kreativě a učení**. Předpoklad známý z výrobních a montážních procesů, který říká, že do procesu musíme zakódovat detailní znalost, aby pracovníci produkovali očekávaný výstup, v této abstraktní a komplexní oblasti, kterou je práce s IT systémy, nefunguje. Proces popisující veškeré aktivity, který nedává prostor pro chybu, nedává prostor ani pro kreativitu a omezuje příležitost k učení (kolik z vás se setkalo buď s procesem či příkazem sdělujícím: „*Teď máš dvě hodiny na to být kreativní a přinést lepší řešení...?*“). Právě učení a kreativita je ale to, co odlišuje procesy tvorby a provozu IT systémů od procesů montážních. Detailní preskriptivní proces je také těžké denně měnit z důvodu adaptace na rychle se měnící podmínky trhu a znalostní ekonomiky dneška. Před několika dekádami a snad ještě i jednotkami let byl ustálený detailní proces předpokladem úspěchu. V dnešní turbulentní době musí organizace rychle reagovat na měnící se požadavky a potřeby zákazníků a trhu samotného. V neposlední řadě takové procesy brání učení tím, že mají dlouhou dobu odezvy přinášející zpětnou vazbu (produkt až na konci vývoje, komentář zákazníka v momentě, kdy už je těžké něco změnit, apod.). Člověk se učí a zlepšuje okamžitou zpětnou vazbou. Proto si například vývojáři používající vývojářské (*unit*) testy, tento přístup tak chválí, protože dostanou odpověď na kvalitu svého kódu v řádu sekund. Chyba, kterou by opakovali neustále v jakékoli napsané komponentě, může být odstraněna ihned.

Problémem organizací je také **sub-optimalizace v rámci organizačních jednotek** vystavěná na domněnce, že výsledný výstup bude správný a nejlepší (firma bude úspěšná), jen když bude správný a nejlepší výstup každé jednotky (zisková a prosperující bude každá jednotka). Proto dekomponujeme cíle jednotlivých organizačních jednotek a očekáváme, že přispějí k celkovému cíli. Místo spolupráce však dostaneme soupeřící prostředí, které neprodukuje to, co jsme očekávali. Příkladem může být plně ziskové a fungující vývojové oddělení doručující kód rychle bez potřebného testování a následování dobrých praktik, na což doplatí oddělení provozu a údržby, které bude přetížené a bude muset odstraňovat problémy zanesené při nekvalitním vývoji. Jaký je navíc smysl maximální ziskovosti oddělení údržby? Neměli bychom se spíše snažit vytvářet, dodávat a provozovat maximálně kvalitní produkt či službu přinášející hodnotu a obsahující minimum chyb? Teorie

systémů [FJ93], [Se09] nám odhaluje, že výsledný systém není pouhým složením jeho částí, ale spíše výsledkem interakcí těchto částí. Úspěch systému tedy závisí na tom, jak dobře spolu jednotlivé části kooperují. Teorie omezení [Sc99], [Go1], [Go2], [Go3], systémové myšlení [Se09] a štíhlá (*Lean*) filozofie [Po06], [Po09] jasně ukazují, že pro úspěch celku (firmy) je třeba odstraňovat úzká místa systému a optimalizovat celek.

Tato kniha vznikla jako reakce na výše popsané problémy. Cílem knihy není přinést teoretický postup, který je vzdálený od problémů v realitě, ani znovu vynalézat již vynalezené. Naším cílem je naopak poskytnout poznatky a předat zkušenost s agilním a „štíhlým“ (ve smyslu anglického *Lean*) způsobem provozu a údržby IT systémů (komplexních informačních systémů a IT služeb provozovaných interně či třetími stranami). Agilní a štíhlé přístupy jsou ověřeným přístupem k tvorbě softwaru. O jejich použití v kontextu provozu, podpory a údržby informačních systémů, respektive IT služeb mnoho literatury nenalezneme. Většina zmínek je o jejich nízké aplikovatelnosti v této oblasti či o aplikaci konkrétních metod jako je Extrémní programování nebo Scrum. Je však pravdou, že pokud organizace vyvíjí software agilně, pak její proces již v sobě obsahuje i údržbu. Kniha, kterou držíte nyní v rukou, má za cíl prezentovat holistický přístup k agilnímu a štíhlému provozu a údržbě, který je postaven na zkušenostech z denní praxe. Holistický přístup musí pokrývat nejen provoz, podporu a údržbu, ale nutně musí být zaměřen i na proces vývoje IS a IT služeb, organizační strukturu a řízení organizace či způsob uzavírání a formu kontraktů. **Zaměření pouze na provoz, podporu a údržbu IS a IT služeb by byl sub-optimalizací jen jedné části celku.** To je důvod, proč v celé knize zmiňujeme návaznost na prodej, proces vývoje či celkovou organizaci a způsob vedení firmy. Postupy jsou popsány ve formě principů, praktik i konkrétních doporučení a příkladů z praxe.

Kniha je zaměřena na podporu činnosti pro všechny pracovníky provozu a údržby, vývoje i obchodu, na manažery i výkonné pracovníky:

- pracovníky (interního či externího) Service Desku na všech úrovních;
- vývojáře opravující chyby a implementující změny;
- testery či test manažery;
- specialisty sestavující pravidelné verze a výsledný produkt;
- obchodníky a týmy připravující nabídky pro výběrová řízení;
- a také (či spíše především) manažery pomáhající těmto lidem (manažery pro styk se zákazníky, projektové manažery či manažery služeb, liniové manažery, dokonce i CEO, CIO).

Běžná námítka, kterou můžeme slyšet při představení principů agilního a štíhlého provozu a údržby, je následující: „*Toto není nic nového, vždyť už je všechno popsáno ve standardech a existujících procesech.*“ Potom je nasnadě ptát se:

- Proč je stále takový problém s chybovým softwarem a neúspěšnými projekty (kterých je více než polovina, viz [SG])?
- Proč se zvyšuje cena provozu a údržby mnohdy dosahující až 90 % celkových IT rozpočtů firem (vlastní zkušenost z velkých mezinárodních společností a empirické výzkumy [Ga08IT], [Ga09IT], [Ho08], [Bo81])?
- Proč je velká fluktuace a demotivace lidí v týmech podpory a údržby, když všechno funguje, jak má a je přesně popsáno ve standardech?

Odpovědí je několik. Jednou z nich jsou procesy nerespektující lidskou přirozenost, tj. nepochopení specifik jednotlivce, potřeby zpětné vazby, prostoru pro kreativitu a učení. Důsledkem je návrh firemních systémů (procesů) jdoucích proti lidské přirozenosti, nefungující učení či chybějící napojení služeb a projektů na podnikové cíle zákazníka. Proto chceme popsat zkušenost s procesně orientovanými frameworky, standardy a metodami, které tyto aspekty opomíjí, a uvést **agilní a štíhlý provoz, podporu a údržbu IT systémů** (IT služeb a informačních systémů) tak, jak jej používáme

a jak jsme si ověřili jeho účinnost v praxi na projektech a službách v České republice, Evropě (převážně Skandinávii) a Indii.

Dnešní procesní přístupy k provozu a údržbě jsou několik (až desítky) let zpožděny za stavem ve vývoji softwaru a s tímto zpožděním evoluci v metodách vývoje softwaru kopírují. Přístupy k vývoji softwaru prošly od ad hoc vývoje v raných dobách tvorby softwaru, přes detailní a rigorózní procesy až po dnešní stav prezentovaný převážně iterativně inkrementálními (resp. agilními) přístupy s důrazem na tým a učení, komunikaci se zákazníkem a delegaci rozhodování. Proč se tedy nepoučít z evoluce přístupů vývoje softwaru a nepřeskočit slepé evoluční větve? O tento krok se snažíme.

Úvodní kapitoly knihy se věnují představení pojmů a současných problémů oblasti provozu a údržby IT systémů, resp. informačních systémů a IT služeb. Mezi hlavní problémy patří rostoucí procento nákladů na provoz a údržbu, které ubírají z financí na inovace a nové projekty. Součástí je také představení současného stavu na poli metod, standardů a metodik pro provoz a údržbu softwaru, IS a IT služeb (**zajímá nás tedy hlavně způsob práce a řízení v této oblasti**). Úspěšnost těchto metod je diskutabilní, proto kapitola přináší empirické důkazy z akademické i industriální sféry a diskutuje možná řešení založená na agilních a štihlých principech.

Jádrem knihy jsou principy agilního a štihlého provozu a údržby, které se ale ve smyslu systémového myšlení nezaměřují jen na provoz, podporu a údržbu, ale integrují tyto činnosti do celého řetězce doručování hodnoty zákazníkovi, tj. do vývoje, obchodu i řízení firem. Agilní principy představují základní filozofii přístupu k provozu a údržbě, která pokrývá jak provozní, tak také řídicí činnosti organizace. Praktiky pomáhají aplikovat tuto filozofii do denních činností v řízení, provozu a údržbě. Pro snazší pochopení a implementaci dané filozofie a praktik se kniha snaží přinést množství příkladů z praxe a popsat těžkosti a návody k řešení reálných situací, se kterými jsme se setkali.

Ve třetí části je představen důležitý aspekt agilního a štihlého provozu, údržby a podpory, který je často opomíjen, jedná se o tzv. *soft aspekty* (komunikace, týmová práce, typologie a psychologie lidí) a poznatky kognitivních věd. Většina pracovníků v IT sféře má vzdělání technického či ekonomického směru, díky němuž rozumí a chápe, jak fungují přírodní zákony, matematické důkazy, či pravidla trhu a oběhu peněz, ale znalost lidského chování, pochopení nepředvídaných reakcí, způsobu myšlení či společné spolupráce je z teoretického i praktického pohledu minimální. Avšak právě tyto aspekty jsou při vývoji, provozu a údržbě kritické, jelikož software je tvořen kreativními lidmi a díky své nehmateľné přirozenosti je jeho diskutování a mentální uchopení obtížnější. Do této části je zařazena také diskuse o kontraktech, jelikož jsou počátkem (resp. dalším nezbytným krokem po optimalizaci pracovního postupu týmu) všech projektů a služeb a jelikož nevhodně nastavený kontrakt může významně omezit jakoukoli možnost interní změny či zlepšení.

Moderní přístupy řešící problémy provozu a údržby deklarují a významně používají nástroje nové generace, mezi něž patří například umělá inteligence či 3D modely a světy. Proto jsou některé z nich představeny v závěrečné části knihy. Tyto přístupy však vyžadují významné investice do nástrojů, procesů, lidských kapacit a schopností. Cílem této knihy je snaha o představení principů a technik, které mohou se stávajícími nástroji a lidským personálem zásadně zefektivnit provoz a údržbu informačních systémů či IT služeb. Cílem je tedy **zlepšení poskytovaných služeb a efektivnosti týmu s minimem finančních investic**. O to více je třeba osobní investice každého pracovníka či manažera (změna myšlení a chování, nalezení vůdců uvnitř týmů apod.).

Struktura knihy je následující:

