

Agilní metody řízení projektů

3. aktualizované
vydání



Zuzana Šochová
Edvard Kunc

Agilní metody řízení projektů

Vyšlo také v tištěné verzi

Objednat můžete na
www.albatrosmedia.cz



Zuzana Šochová, Eduard Kunc

Agilní metody řízení projektů – e-kniha
Copyright © Albatros Media a. s., 2024

Všechna práva vyhrazena.
Žádná část této publikace nesmí být rozšiřována
bez písemného souhlasu majitelů práv.


ALBATROS MEDIA

Zuzana Šochová, Eduard Kunc

Agilní metody řízení projektů



Zuzana Šochová, Eduard Kunce

Agilní metody řízení projektů

Vydalo nakladatelství Computer Press v Brně roku 2024 ve společnosti Albatros Media a. s.
se sídlem 5. května 22, Praha 4.

Číslo publikace 43 610.

Jazyková korektura: Marie Schreinerová

Odpovědný redaktor: Roman Bureš

Technický redaktor: Jiří Matoušek

3. vydání

© Zuzana Šochová, 2024

© Eduard Kunce, 2024

ISBN tištěné verze 978-80-251-5107-5

ISBN e-knihy 978-80-251-5108-2 (1. zveřejnění, 2024) (ePDF)

e-shop: www.albatrosmedia.cz

Cena uvedená výrobcem představuje nezávaznou doporučenou spotřebitelskou cenu.



Obsah

Úvod	9
 Zpětná vazba od čtenářů.....	11
 Errata.....	11
Část I – Filozofie agilních a lean metod.....	13
 Agilní a lean metody.....	15
Co si představíte pod slovem „agilní“?	15
Manifest agilního vývoje softwaru	15
Co je to „lean“?	25
Co je to Scrum?	26
Co je to Kanban?	26
Co je to Scaling (škálování)?	26
 Agilní transformace	29
Vývoj softwaru	29
Proč vůbec něco měnit	31
Jaké jsou nejčastější důvody pro přechod na agilní metody?.....	31
Business Agility	35
Agilní organizace a agilní leadership	35

Část II – Popis metod, procesů, praktik a artefaktů	37
Agilní slovníček	38
Lidi a vztahy	43
Scrum Master	43
Product Owner.....	47
Self-managed tým	49
Multifunkční tým.....	51
Scrum tým	53
Zákazník.....	55
Role manažera	57
Role projektového manažera	58
Praktiky a artefakty	61
Sprint Goal.....	61
Prioritizace.....	63
Product Backlog Item	67
Sprint.....	69
Product Backlog	73
Sprint Backlog	75
User Story.....	77
INVEST kritéria	79
Epic a Témata	83
Definition of Done	84
Dobrá vizualizace	87

Scrum tabule	87
Kanban tabule	89
Kdy potřebujeme nástroj?	89
Ohodnocení	93
Relativní jednotky versus odhad času	95
Odhadování ve Scrumu	97
Planning Poker	99
Rychlost	103
Eventy a aktivity	107
Standup	107
Daily Scrum	108
Retrospektiva	111
Další formáty Retrospektivy	114
Backlog Refinement	119
Backlog Grooming	119
Sprint Planning	121
Sprint Review	125
Agilní programování – artefakty	127
Pair Programming	127
Mob Programming	128
Review	128
Sdílený kód	128
Coding Standard	129

Jednoduchý design	130
Continuous Integration.....	131
Test Driven Development – TDD	132
Refactoring	133
Časté releasy.....	134
Žádné přesčasy	134
Společný cíl.....	134
Technický dluh.....	135
Scrum 1-1	139
Příprava Scrumu	141
Scrum pro více týmů – Scaling	143
Scrum of Scrums	144
Komunity	145
Extreme Programming 1-1	147
Kanban 1-1	151
Jak funguje tým	155
Tuckman’s Group Development Model.....	155
Pět důvodů, proč týmy nefungují	158
Smlouvy a formy spolupráce	163
Fixed Time, Fixed Price.....	163
Agilní smlouva	164
Otázky na závěr	167

Část III – Implementace, transformace, změna	171
Průvodce agilní změnou	173
Proč agilně.....	173
Bude to zig-zag	175
Co máme dělat, když...	177
Co máme dělat, když jsme malá firma?	177
Co máme dělat, když jsme středně velká firma?	178
Co máme dělat, když jsme velká firma?	178
Co máme dělat, když děláme jen migraci?.....	181
Co máme dělat, když jsme projektově organizovaná firma?	183
Co máme dělat, když máme distribuovaný tým?.....	184
Jak stavět Product Backlog.....	187
Product Discovery.....	187
Story Mapping a Project Chartering.....	189
„Best Practices“ Scrumu pro software prostředí	192
Nejčastější nepochopení Scrumu	194
Agilní organizace	197
Ohodnocování a metriky.....	201
 Část IV – Příklady.....	 203
Průvodce agilní transformací	
– co budete při zavádění Scrumu potřebovat	205
Ukázka backlogu	207

Část V

Případové studie	209
Jak změnit styl práce	211
Menší firma – cesta k agilní organizaci	211
Kontext	211
První fáze: Ostrůvky (2005–2006)	211
Druhá fáze: Agilní týmy (2006–2008)	212
Třetí fáze: Agilní organizace (2008–2010)	213
„Spotify Model“	215
Mezinárodní korporace	216
Ambiciózní startup	217
Závěr	219
O autorech	220
Zuzana Šochová	220
Eduard Kunce	221
Ostatní kontakty	222
Agilní asociace	222
Agile Prague Conference	222
Reference	224

Úvod

Dostává se vám do ruky publikace o agilních metodách. Snažili jsme se čtivou formou popsat všechna možná zákoutí a nástrahy, které vás při přechodu na agilní metody mohou potkat. Kniha nepopisuje jen teorii, co to agilní metody jsou, jak funguje Scrum a Kanban a jak by jednotlivé artefakty těchto procesů měly správně vypadat, ale primárně se snaží vysvětlit filozofii agilního přístupu a zaměřit se na objasnění, proč jednotlivé metody fungují. Součástí textu jsou praktické příklady a doporučení co dělat a čeho se naopak vyvarovat.

Kniha se zabývá agilními metodikami vývoje softwaru v různých typech společností, od malých firem až po nadnárodní korporace. První část knihy, **Filozofie agilních a lean metod**, popisuje základní směry agilního a lean světa a proč vlastně vzniká potřeba agilní transformace. Druhá část, **Popis metod, procesů, praktik a artefaktů**, začíná slovníčkem pojmů, pokračuje

čuje detailním popisem jednotlivých rolí, praktik, artefaktů, eventů a aktivit a končí popisem jednotlivých frameworků. Třetí část, **Implementace, transformace, změna**, je průvodcem agilní změnou a obsahuje praktické rady pro konkrétní situace. Čtvrtá část – **Příklady** – obsahuje seznamy toho, co budete potřebovat změnit, a konkrétní příklady jak začít. Druhé, rozšířené a upravené vydání knihu doplňuje o pátou část, **Případové studie** z agilních transformací v různých organizacích, a také upravuje doporučené praktiky vzhledem k současnému state-of-the-art agilního světa.

V posledních 20 letech se agilní metody staly ve světě velmi populární. Používají je velké nadnárodní korporace jako Amazon, Salesforce, Oracle či Google, najdete je na univerzitách, v telekomunikacích, v bankách a pojišťovnách, v armádě, aerolinkách, automobilovém průmyslu, medicínských firmách a samozřejmě

i v mnoha malých či středních firmách. V České republice se dostávají do povědomí firem až v posledních několika letech, ale být agilní, používat Scrum či Kanban již dávno není výjimkou a agilní metody se tak stávají na českém trhu běžnou metodikou.

Kniha je primárně určena pro lidi pracující v ICT, a to jak pro softwarové vývojáře, testery, designéry, architekty a analytiky, tak i projektové a produktové manažery, vedoucí oddělení, manažery i ředitele či vlastníky takových organizací. Přestože agilní metody byly primárně definované pro zlepšení a zefektivnění vývoje softwaru, používají se v podstatě kdekoli, kde je třeba řídit projekty, organizovat týmy lidí a kde je výsledný produkt komplexní a pro jeho úspěch je vyžadována kreativita, flexibilita, schopnost reagovat na změny, např. marketingové či reklamní agentury, konzultantské společnosti, HW firmy. Máte-li na starosti nějaký projekt nebo tým lidí, určitě je tato kniha pro vás.

A kladete-li si hned v začátku otázku, co můžete spolu se zavedením agilních metod očekávat, tak vyšší flexibilitu a schopnost reagovat na změny, vyšší efektivitu, kvalitnější produkt, větší předvídatelnost termínu dodání a vyšší spokojenost zákazníka s vaším produktem či službou a v neposlední řadě motivovaný tým, který se spolupodílí na produktu a jeho inovacích. Tedy jednoduše řečeno být úspěšnější.

Autoři vycházejí z vlastní zkušenosti s nasazováním agilních metodik v různých prostředích firem, od malých startupů až po mezinárodní korporace. Ze svých bohatých zkušeností autoři vybrali vše podstatné a srozumitelnou formou předkládají čtenáři návod, jak postupovat, od prvních krůčků až po kompletní agilní transformaci celé společnosti.

Zpětná vazba od čtenářů

Nakladatelství a vydavatelství Computer Press, které pro vás tuto knihu připravilo, stojí o zpětnou vazbu a bude na vaše podněty a dotazy reagovat. Můžete se obrátit na následující adresy:

Computer Press

Albatros Media a.s., pobočka Brno

IBC

Příkop 4

602 00 Brno

nebo

sefredaktor.pc@albatrosmedia.cz

Computer Press neposkytuje rady ani jakýkoli servis pro aplikace třetích stran. Pokud budete mít dotaz k programu, obraťte se prosím na jeho tvůrce.

Errata

Přestože jsme udělali maximum pro to, abychom zajistili přesnost a správnost obsahu, chybám se úplně vyhnout nelze. Pokud v některé z našich knih najdete chybu, budeme rádi, pokud nám ji oznámíte. Ostatní uživatelé tak můžete ušetřit frustrace a nám pomůžete zlepšit následující vydání této knihy.

Veškerá existující errata najdete na stránce knihy na www.albatrosmedia.cz po klepnutí na odkaz Soubory ke stažení. (Nejsou-li žádná errata zatím k dispozici, není odkaz Soubory ke stažení dostupný.)

Část I

Filozofie agilních a lean metod



Agilní a lean metody

Co si představíte pod slovem „agilní“?

Agilní je dynamický, rychlý, interaktivní, přizpůsobivý, iterativní, zábavný, hravý, rychle reagující na změnu... a jistě vymyslíte spoustu dalších synonym. Je to jiný způsob života, upřednostňující jiné hodnoty. Reálný výsledek před striktními procesy, změnu před předem naplánovaným. Být agilní znamená žít agilní filozofií, je to odlišná firemní kultura a nálada. Ale jestli čekáte kuchařku, jak se stát agilní, budete zklamaní. A jestli očekáváte nějaký model či framework s několika stupni agility, nebo dokonce checklist, tak vás bohužel zklame. Nic takového nehledejte, protože to ani neexistuje, ač mnoho certifikovaných hlav tvrdí pravý opak. Žádný certifikát agility vám v pochopení nepomůže, maximálně může nastartovat proces přeměny. Ale agilní musíte být, agilně musíte myslet, agilně se chovat.

Ale nebojte se, není to zas tak neuchopitelné, jak se na první pohled zdá. Agile je o spolupráci a komunikaci a připravenosti na změnu. O tom, že zásadně děláme to, co má v danou chvíli smysl, a děláme to tak, jak nejlépe umíme. Agile sice není žádný striktní proces, ale není to ani žádný chaos. Má svá jasná pravidla. Dalo by se říct, že definuje hranice a vytyčuje menší hřiště, v jejichž rámci si týmy mohou stanovovat svá vlastní

pravidla hry, tak aby se jim dobře pracovalo a byli všichni co nejvíce produktivní, efektivní a dodali kvalitní produkt v co nejkratším čase. Takový tým je zaměřený na Business Value, tedy hodnotu pro zákazníka. Na to, jak optimalizovat funkcionalitu tak, aby zákazník byl maximálně spokojený a dostal za vynaložené prostředky to, co opravdu potřebuje a může používat.

Základním stavebním kamenem je – jak jinak – manifest; v tomto případě přímo agilní manifest, který shrnuje ve čtyřech bodech, co to vlastně znamená být agilní.

Manifest agilního vývoje softwaru

<http://agilemanifesto.org>

Objevujeme lepší způsoby vývoje softwaru tím, že jej tvoříme a pomáháme při jeho tvorbě ostatním. Při této práci jsme dospěli k těmto hodnotám:

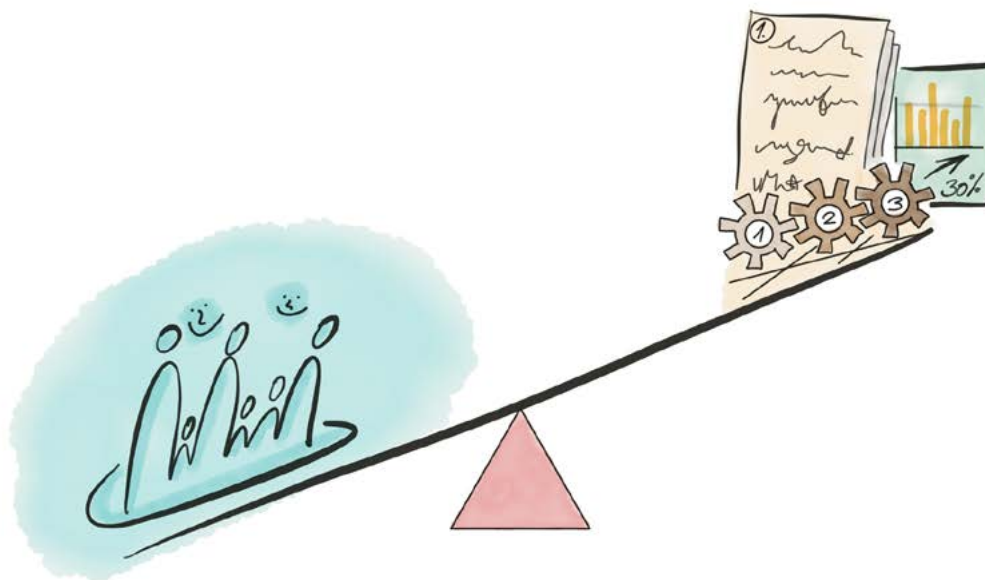
- ☐ Jednotlivci a interakce před procesy a nástroji
- ☐ Fungující software před vyčerpávající dokumentací
- ☐ Spolupráce se zákazníkem před vyjednáváním o smlouvě
- ☐ Reagování na změny před dodržováním plánu

Jakkoliv jsou body napravo hodnotné, bodů nalevo si ceníme 4ice.

Manifest Agilního vývoje softwaru vznikl v roce 2001, jako reakce na malou úspěšnost projektů. Skupina sedmnácti softwarových inženýrů se dala dohromady a přemýšleli, jak by mohli pomoci ostatním mít projekty úspěšnější. Co by jim mohli poradit. Tak vznikl Manifest Agilního vývoje softwaru. Je tedy zaměřený

na prostředí vývoje softwaru. Neznamená to však že se agilní metody hodí jenom v softwarovém vývoji. Dnes se používají v podstatě ve všech odvětvích – zdravotnictví, telekomunikace, finančnictví, výroba, marketing, obchod, nebo třeba školství.

Pojďme se společně chvíli zamyslet nad tím, co nám vlastně manifest říká.



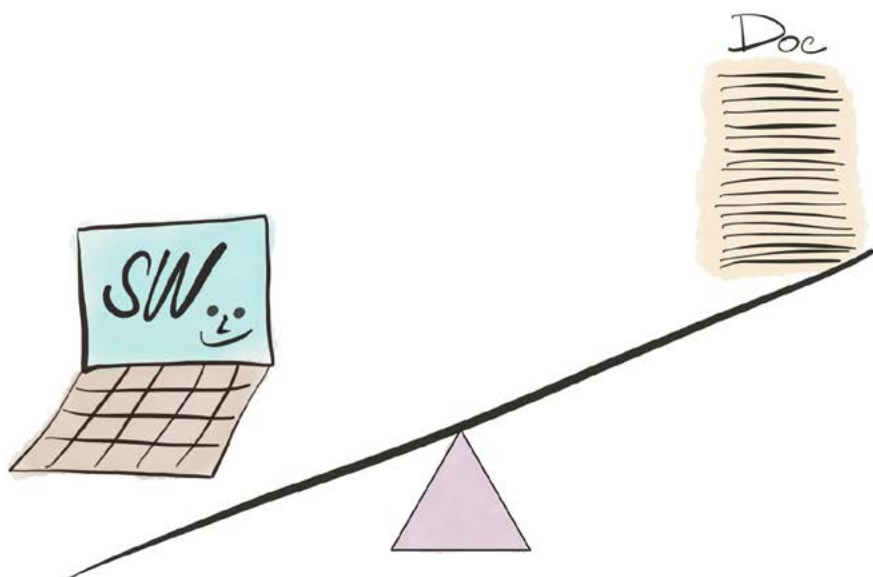
Jednotlivci a interakce před procesy a nástroji

Je známým faktem, že spolupracující týmy mají lepší výsledky než skupiny individuálně pracujících jednotlivců. Procesy a nástroje jim pomáhají dosáhnout výsledků, ale nejsou pro jejich úspěch nijak klíčové. Pojďme se podívat na dva odlišné týmy. V prvním máme 10 profesionálů, kteří si vytvořili vlastní nástroje a pracují v silně kolaborativním prostředí, ve druhém máme 10 vývojářů pracujících v přesně definovaném procesním prostředí s nejlepšími nakoupenými nástroji, ale bez interakcí a spolupráce.

Co myslíte, který z těchto týmů vytvoří lepší software v kratším čase? Myslím, že se shodneme na tom, že první

tým má větší šanci uspět, protože vzájemná spolupráce a komunikace mnohokrát převáží přesně definované procesy a nástroje. Ostatně podívejme se na úspěšné startupy, kde moc procesů nenajdeme, zato všichni překypují spoluprací a komunikací. No, a hlupák i s nejlepším nástrojem je zkrátka pořád hlupák.

Na druhou stranu, manifest neříká nic o tom, že procesy a dohody by neměly existovat ani že by týmy měly pracovat zcela bez nástrojů. Jen by měly mít možnost si nástroje vybrat a používat jen ty, které jim opravdu pomáhají v dosažení kvalitního výsledku.



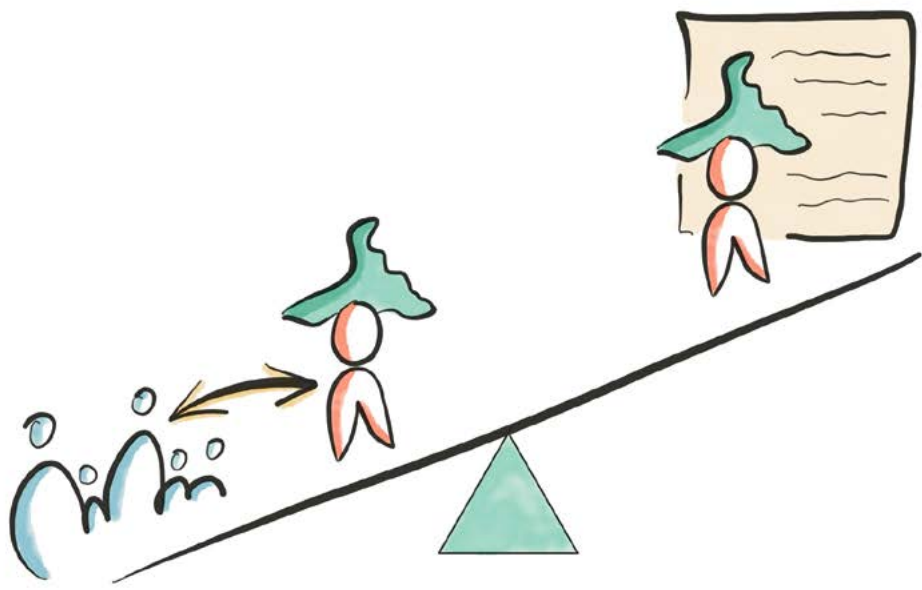
Fungující software před vyčerpávající dokumentací

Když se zeptáte zákazníků, zda by si koupili dokument popisující váš plánovaný produkt s architektonickou dokumentací, nebo ten samý produkt s minimální dokumentací, co si vyberou? Nebo praktičtější příklad: Když si jdete koupit nový mobilní telefon, vyberete si pouze podle technické specifikace na papíře, nebo pro vás bude důležité si vyzkoušet i vlastní telefon před tím, než si ho koupíte? A když si pořídíte nový televizor, přečtete si nejdříve manuál, nebo chcete vyzkoušet, jak funguje, a manuál čtete až na posledním místě? Lidé (ano, i zákazníci jsou lidé) zkrátka preferují praktické seznamování s produktem před teoretickým.

Mnoho softwarových firem na toto zapomíná a chce ohromit zákazníka množstvím dokumentace, ale když dojde na vlastní demo, výsledek často dokumentaci odporuje a zákazník je z „funkčního“ softwaru poněkud překvapený. Dokumentace **je** důležitá, ale neměla by převážet nad vlastním produktem; měla by primárně sloužit jako reference pro oblasti, které nejsou intuitivní a snadno pochopitelné. A takových oblastí by mělo být minimum.

Interní dokumentace se často píše pro budoucí týmy, aby se znalost architektury produktu neztratila. To samo o sobě není nijak špatný nápad, ale už jste někdy zkoušeli něco dohledat v deseti úrovních dokumentů, každý o 1000+ stranách? Vyčerpávající. A když už si dáte tu práci a prohledáte tu kupu slov, v devíti z deseti případů zjistíte, že daná věc v dokumentaci chybí či je v poslední verzi již zcela jinak. Interní dokumentace by měla být stručná a postihnout klíčové informace. Ostatní je obvykle efektivnější dokumentovat přímo v kódu.

Posledním případem je dokumentace funkcionalit v průběhu vývoje, tu můžete v podstatě úplně odstranit a nahradit dobrou komunikací mezi analytiky, testery a vývojáři. Ti se pak sami domluví, co je třeba zdokumentovat pro „budoucí generace“ vývojářů a testerů a v jaké podobě. Na závěr zdůrazníme, že nedoporučujeme přestat dokumentovat, ale jen dokumentaci omezit na minimum, aby poměr mezi námahou a časem, jež investujete do psaní dokumentace, odpovídal hodnotě, kterou její zákazníci, tedy všichni, kdo ji kdy čtou, z takovéto dokumentace načerpají.



Spolupráce se zákazníkem před vyjednáváním o smlouvě

Proč vlastně vytváříme nějaký produkt? Chceme ho prodat, že? Když stavíme produkty pro zákazníky, je dobrý nápad se jich zeptat, co vlastně chtějí koupit. Jistě, zákazník mnohdy neví, co vlastně chce, a často mění názor, ale dlouhodobou spoluprací můžeme zákazníka „vychovat“ a společně vytvořit spolupracující tým, který zajistí jak úspěch náš, tak úspěch našeho klienta. Je lepší se se zákazníkem dohodnout a spolupracovat, než se potkávat pouze u soudu a komunikovat přes právní zástupce.

Smlouvy **jsou** důležité, ale neměly by být prostředkem nahrazujícím spolupráci a komunikaci. Ostatně není od věci se už při jejím podpisu zamyslet nad tím,

co je opravdu reálné, že se při spolupráci stane – tedy že zákazník možná ne vždy ví přesně, co chce, a že se stejně bude měnit scope – a smlouvu se tak pokusit přiblížit realitě.

Spokojený zákazník, který dostal to, co potřebuje, vás doporučí dalším firmám, zatímco zákazník, který sice má to, co si objednal a ve smlouvě podepsal, ale nijak mu to v jeho misi nepomohlo, se příště podívá po jiném dodavateli. Smlouvy určitě pište, ale ani sebelepší smlouva vás neochrání před změnami, co přijdou. Obvykle to stejně bez ohledu na smlouvu skončí kompromisem, kdy po měsících hádek o chyby a change requesty jich dodavatel polovinu dodělá jako opravu na vlastní náklady a polovinu jako novou funkčnost za extra peníze od zákazníka. Vzájemné spokojenosti a dobrým vztahům to nijak nepomůže.