

Zuzana Šochová, Eduard Kunc

AGILNÍ METODY ŘÍZENÍ PROJEKTŮ



computer
press

Zuzana Šochová, Eduard Kunc

Agilní metody řízení projektů

**Computer Press
Brno
2014**

Agilní metody řízení projektů

Zuzana Šochová, Eduard Kunc

Obálka: Martin Sodomka

Odpovědný redaktor: Martin Herodek

Technický redaktor: Jiří Matoušek

Objednávky knih:

<http://knihy.cpress.cz>

www.albatrosmedia.cz

eshop@albatrosmedia.cz

bezplatná linka 800 555 513

ISBN 978-80-251-4194-6

Vydalo nakladatelství Computer Press v Brně roku 2014 ve společnosti Albatros Media a. s. se sídlem Na Pankráci 30, Praha 4. Číslo publikace 18 468.

© Albatros Media a. s. Všechna práva vyhrazena. Žádná část této publikace nesmí být kopírována a rozmnožována za účelem rozšiřování v jakékoli formě či jakýmkoli způsobem bez písemného souhlasu vydavatele.

1. vydání


ALBATROS MEDIA a.s.

Obsah

Úvod	9
Zpětná vazba od čtenářů	10
Errata	10

ČÁST I

FILOZOFIE AGILNÍCH A LEAN METOD

KAPITOLA 1

Agilní a Lean metody	13
Co si představíte pod slovem „Agilní“?	13
Manifest Agilního vývoje software	14
Jednotlivci a interakce před procesy a nástroji	14
Fungující software před vyčerpávající dokumentací	15
Spolupráce se zákazníkem před vyjednáváním o smlouvě	16
Reagování na změny před dodržováním plánu	17
Co je to „lean“?	18

KAPITOLA 2

Transformace na agilní metody	21
Vývoj Software	21
Proč vůbec něco měnit	22
Jaké jsou nejčastější důvody pro přechod na agilní metody?	23

ČÁST II

POPIS METOD, PROCESŮ, PRAKTIK A ARTEFAKTŮ

KAPITOLA 3

Agilní slovníček	27
-------------------------	-----------

KAPITOLA 4

Role	31
Scrum Master	31
Product Owner	33
Self-organized tým	34
Scrum tým	35
Zákazník	37

Product Owner Proxy	39
Role manažera	39
Role projektového manažera	41

KAPITOLA 5

Artefakty	43
Sprint	43
Product Backlog	45
Sprint Backlog	47
User Story	48
INVEST kritéria	50
Epic a Super User Story	52
Akceptační kritéria	53
Done kritéria	53
Dobrá vizualizace	54
Scrum tabule	55
Kanban Tabule	56
Kdy potřebujeme nástroj?	56
Ohodnocení	58
Relativní jednotky versus odhad času	60
Planning Poker	61
Rychlost	63
Burndown graf	64

KAPITOLA 6

Meetingy	67
Standup / Scrum Meeting	67
Retrospektiva	70
Jak začít?	71
Retrospektiva s lepítky	72
Časová osa	73
Další metody, jak retrospektivu občerstvit	75
Priority a Pre-planning	76
Plánování – Planning	79
Sprint Review	81
Backlog Grooming	83

KAPITOLA 7

Agilní programování – artefakty	85
Pair Programming	85
Review	86
Sdílený kód	86

Coding Standard	87
Jednoduchý design	88
Continuous Integration	89
Test Driven Development – TDD	90
Refaktorování	90
Časté releasy	91
Žádné přesčasy	91
Společný cíl	92
Technický dluh	92

KAPITOLA 8

Scrum 1-1 a škálovatelnost Scrumu **95**

Příprava Scrumu – Sprint 0	98
Scrum pro více týmů	98
Scrum of Scrums a další synchronizační meetingy	99

KAPITOLA 9

Extreme Programming 1-1 **101**

KAPITOLA 10

Kanban 1-1 **105**

KAPITOLA 11

Jak funguje tým **107**

Tuckman's Group Development Model	107
Pět důvodů proč týmy nefungují	110
Důvěra	111
Strach z konfliktu	111
Nedostatek ochoty k závazku	111
Jak funguje zodpovědnost	112
Neochota převzít odpovědnost	113

KAPITOLA 12

Smlouvy a formy spolupráce **115**

Fixed Time, Fixed Price	116
Agilní smlouva	117

KAPITOLA 13

Otázky na závěr **119**

Jsou agilní metody a Scrum vhodné pouze pro malé firmy?	119
Má Scrum smysl jen na projektech, kde se často mění zadání?	119
Je vhodné zavádět jednotlivé metody postupně?	120

Pro které firmy agilní přístup není vhodný?	120
Musí tým sedět společně v jedné místnosti?	120
Musí být jednotliví členové v týmu alokováni na plný úvazek?	121
Jak moc nám agilní metody můžou pomoci?	121

ČÁST III

IMPLEMENTACE, TRANSFORMACE, ZMĚNA

KAPITOLA 14

Průvodce agilní změnou	125
Proč agilně	125
Bude to Zig-Zag	127

KAPITOLA 15

Co mám dělat, když...	129
Co mám dělat, když jsem malá firma?	129
Co mám dělat, když jsem středně velká firma?	130
Co mám dělat, když jsem velká firma?	131
Top-down	131
Bottom-up	132
Co mám dělat, když děláme jen migraci?	134
Co mám dělat, když jsme projektově organizovaná firma?	135
Co mám dělat, když máme distribuovaný tým?	136

KAPITOLA 16

Jak stavět Product Backlog	139
Product Discovery	139
Story Mapping a Project Chartering	141

KAPITOLA 17

„Best Practices“ Scrumu	147
--------------------------------	------------

KAPITOLA 18

Agilní organizace a Tulming metoda	151
Tulming metodika	153

KAPITOLA 19

Ohodnocování a metriky	155
-------------------------------	------------

ČÁST IV
PŘÍKLADY

KAPITOLA 20

Průvodce agilní transformací – co budete při zavádění Scrumu potřebovat	159
Role	159
Artefakty	160
Praktiky	161

KAPITOLA 21

Ukázka Backlogu	163
------------------------	------------

Závěr	165
--------------	------------

O Autorech	167
-------------------	------------

Zuzana Šochová	167
----------------	-----

Eduard Kunce	168
--------------	-----

Ostatní kontakty:	168
--------------------------	------------

Agilní Asociace	168
-----------------	-----

Agile Prague Conference	169
-------------------------	-----

Rejstřík	171
-----------------	------------

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 pouze to, 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 vysvětlení, 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 společnostech, od malých firem až po nadnárodní korporace. V první části knihy jsou popsány základní stavební prvky agilního vývoje, od vlastní definice procesu až po ukázky z praxe. Čtenář se dozví všechny potřebné informace k adopci agilních metodik, kulturní odlišnosti agilních firem a další základní stavební kameny nutné pro úspěšnou adopci agilních principů. Kniha dále obsahuje podrobný popis rolí, které se typicky vyskytují v agilně řízených firmách. Vedle těchto popisů obsahuje kniha i „kuchařky“, resp. doporučené postupy adopce agilního řízení pro různé velikosti a typy firem.

V posledních 10 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 a 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. v marketingových či reklamních agenturách, konzultantských společnostech nebo hardwarových firmách). Máte-li na starosti nějaký projekt nebo tým lidí, určitě je tato kniha pro vás.

A kladete si hned v začátku otázku, co můžete spolu se zavedením agilních metod očekávat? Určitě 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 spolu-podílí 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 pomoci nám zlepšit následující vydání této knihy.

Veškerá existující errata zobrazíte na adrese <http://knihy.cpress.cz/K2140> po klepnutí na odkaz Soubory ke stažení.

ČÁST



FILOZOFIE AGILNÍCH A LEAN METOD

- **KAPITOLA 1** – Agilní a Lean metody
- **KAPITOLA 2** – Transformace na Agilní metody

Agilní a Lean metody

V této kapitole:

- Co si představíte pod slovem „agilní“?
- Manifest agilního vývoje softwaru
- Co je to „lean“?

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 filosofií. Přináší to odlišnou firemní kulturu a náladu. 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 zklameme. 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. Děláme zásadně to, co má v danou chvíli smysl, a děláme to tak, jak nejlépe umíme. Nejde sice o 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 byly 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

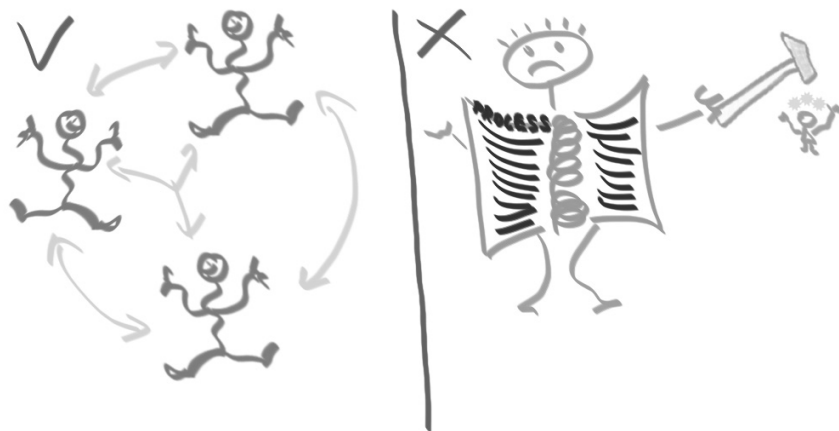
Jakkoliv jsou body napravo hodnotné, bodů nalevo si ceníme více. Pojďme se společně chvíli zamyslet nad tím, co nám vlastně manifest říká.

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, kteří pracují 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. Hlupák i s nejlepším nástrojem je zkrátka pořád hlupák.

Na druhou stranu, manifest neříká nic o tom, že procesy a dohody my 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 popis plánovaného produktu 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 bude pro vás důležité si vyzkoušet i vlastní telefon před tím, než si ho koupíte? A když si koupíte 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.

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, z nich každý má aspoň 1 000 stran? 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.

Poslední případ je dokumentace funkcionalit v průběhu vývoje, kterou 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 připomeneme, že nedoporučujeme přestat dokumentovat, ale jen dokumentaci omezit na minimum, aby poměr mezi námahou a časem, který investujete do psaní dokumentace, odpovídal hodnotě, kterou její zákazníci, tedy všichni, kteří 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 zákazníka. 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. Je důležité se už při jejím podpisu zamyslet nad tím, co se při spolupráci může stát. Je třeba si uvědomit, že zákazník možná ne vždy ví přesně, co přesně chce, a že se stejně bude měnit rozsah a smlouvu se 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 píšete, ale ani sebelepší smlouva vás neochrání před změnami, které přijdou. Obvykle to stejně bez ohledu na smlouvu skončí kompromisem, kdy po měsících hádek o chyby a požadavky na změny 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.



Reagování na změny před dodržováním plánu

Svět se pořád hýbe kupředu a technologie se neustále mění a spolu s nimi se mění i požadavky a problémy našich zákazníků. Ti, jako všichni ostatní, se musí přizpůsobovat trendům na trhu a konkurenci. A my, jako jejich dodavatel, je nemůžeme v těchto změnách brzdit – tedy pokud chceme dále spolupracovat. Představte si, že zákazník přijde se změnou v posledních fázích projektu a ta změna bude důležitá pro jeho přežití na trhu.

Řeknete mu, že se budete držet plánu a kontraktu a změny přijdou až po doručení první verze. A opravdu si myslíte, že další změny budou? Odvážím se tvrdit, že pokud se my nepřizpůsobíme, náš zákazník už si nic dalšího nekoupí, zkrátka proto, že ho konkurence převálcuje. Podobně je to i v méně vyhraněných případech, kdy zákazník prostě jen zjistí, že to, co původně tolik chtěl, vlastně vůbec neřeší jeho problémy a že by naopak potřeboval něco úplně jiného, o čem se předtím vůbec nemluvilo.