

Legendární titul
**SOFTWARE
REQUIREMENTS**
2. vyd. v češtině

Karl E. Wiegiers

dvojnásobný držitel ceny Productivity Award
časopisu Software Development

Požadavky na software

Od zadání
k architektuře
aplikace

Praktické techniky
sběru a zpracování
požadavků
v procesu vývoje

Jak snížit náklady
na údržbu
a podporu

Desítky návodů
zkušeného analytika



C PRESS

Microsoft®

Karl E. Wiegers

Požadavky na software

Computer Press, a.s.
Brno
2008

Požadavky na software

Karl E. Wiegers

Computer Press, a.s., 2008. Vydání první.

Překlad: Tomáš Znamenáček

Odborná korektura: Václav Pergl

Jazyková korektura: Pavel Bubla

Vnitřní úprava: Jiří Matoušek

Sazba: René Kašík

Rejstřík: René Kašík

Obálka: Martin Sodomka

Komentář na zadní straně obálky:

Radek Hylmar

Technická spolupráce: Zuzana Šindlerová,

Tomáš Tůma

Odpovědný redaktor: Radek Hylmar

Technický redaktor: Jiří Matoušek

Produkce: Daniela Nečasová

Copyright 2003 by Microsoft Corporation.

Original English language edition Copyright © 2003 by Karl Wiegers

All rights published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, USA.

Czech Language Edition, published by Computer Press, a.s.

Autorizovaný překlad z originálního anglického vydání Software Requirements, Second Edition.

Originální copyright: © Karl Wiegers, 2003.

Překlad: © Computer Press, a.s., 2008.

Computer Press, a. s.,

Holandská 8, 639 00 Brno

Objednávky knih:

<http://knihy.cpress.cz>

distribuce@cpress.cz

tel.: 800 555 513

ISBN 978-80-251-1877-1

Prodejní kód: K1567

Vydalo nakladatelství Computer Press, a. s., jako svou 2942. publikaci.

© Computer Press, 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.

Stručný obsah

Část I

Podstata a význam požadavků na software

1. Nejdůležitější požadavek na software	25
2. Požadavky z pohledu zákazníka	43
3. Dobré zvyky pro řízení požadavků	55
4. Analytik požadavků	71

Část II

Vývoj požadavků

5. Produktová vize a rozsah projektu	83
6. Jak najít zákazníkův hlas	97
7. Jak nepřeslechnout zákazníkův hlas	113
8. Jak správně pochopit uživatelské požadavky	127
9. Hrajeme podle pravidel	145
10. Dokumentace požadavků	155
11. Jeden obrázek vydá za 1024 slov	177
12. Kvalitativní parametry aneb Víc než funkce	197
13. Snižování rizika prototypováním	213
14. Rozdělení priorit	225
15. Kontrola požadavků	235
16. Problémové typy projektů	255
17. Požadavky versus ostatní práce na projektu	267

Část III

Správa požadavků

18. Principy a techniky pro správu požadavků	281
19. Když se přihodí změny	291
20. Dohledatelnost požadavků	311
21. Nástroje pro správu požadavků	323

Část IV

Řízení požadavků

22.	Zlepšování stávajících procesů	335
23.	Softwarové požadavky a řízení rizik	351

Část V

Přílohy

A.	Dotazník pro posouzení vašich aktuálních procesů	367
B.	Vztah mezi požadavky a modely zlepšování procesů	373
C.	Hledání chyb v požadavcích	379
D.	Ukázková dokumentace požadavků	401

Obsah

O autorovi	15
Úvodem	17
Výhody, které vám tato kniha nabízí	18
Pro koho je tato kniha určena	18
Co v knize najdete	18
Případové studie	19
Od slov k činům	19
Poděkování	21

Část I

Podstata a význam požadavků na software

KAPITOLA 1

Nejdůležitější požadavek na software	25
Definice softwarových požadavků	28
Několik výkladů slova požadavek	28
Typy požadavků	29
Co mezi požadavky nepatří	32
Vývoj a správa požadavků	32
Vývoj požadavků	33
Správa požadavků	33
Každý projekt má své požadavky	35
Když se dobrým lidem přihodí špatná analýza požadavků	36
Nedostatečné zapojení uživatelů	36
Tiché přidávání požadavků	37
Nejednoznačné požadavky	37
Šperkování	38
Příliš malá specifikace	38
Přehlédnuté skupiny uživatelů	38
Nepřesné odhady	38
Výhody kvalitního zpracování požadavků	39
Vlastnosti dobře popsanych požadavků	40
Vlastnosti jednotlivých požadavků	40
Vlastnosti celé specifikace	41
Další kroky	42

KAPITOLA 2

Požadavky z pohledu zákazníka 43

Kdo je zákazník	45
Vztah mezi zákazníkem a vývojáři	46
Seznam základních práv softwarového zákazníka	47
Seznam základních povinností softwarového zákazníka	49
A co podepisování	52
Další kroky	53

KAPITOLA 3

Dobré zvyky pro řízení požadavků 55

Znalosti	57
Sbírání požadavků	58
Analýza požadavků	60
Specifikace požadavků	61
Kontrola požadavků	62
Správa požadavků	63
Řízení projektu	64
Zavádění nových zvyků do praxe	65
Vývoj požadavků jako proces	67
Další kroky	69

KAPITOLA 4

Analytik požadavků 71

Úloha analytika požadavků	71
Analytikovy úkoly	73
Nezbytné analytické dovednosti	75
Nezbytné analytické znalosti	76
Jak se dělá analytik	77
Bývalý uživatel	77
Bývalý vývojář	78
Oborový expert	78
Jak podpořit spolupráci	79
Další kroky	79

Část II

Vývoj požadavků

KAPITOLA 5

Produktová vize a rozsah projektu	83
Definice vize podle podnikatelských požadavků	84
Protichůdné podnikatelské požadavky	85
Podnikatelské požadavky a případy užití	86
Dokumentace vize a rozsahu projektu	86
Podnikatelské požadavky	87
Vize řešení	89
Rozsah a omezení	90
Podnikatelský kontext	91
Kontextový diagram	93
Jak dodržet rozsah projektu	94
Další kroky	95

KAPITOLA 6

Jak najít zákazníkův hlas	97
Zdroje požadavků	98
Třídy uživatelů	99
Jak najít vhodné zástupce uživatelů	102
Produktový šampión	104
Produktový šampión zvenčí	105
Požadavky na produktového šampióna	106
Optimální počet produktových šampiónů	107
Jak produktové šampióny prosadit	108
Na co si dát pozor	108
Kdo bude rozhodovat?	109
Další kroky	110

KAPITOLA 7

Jak nepřeslechnout zákazníkův hlas	113
Sbírání požadavků	114
Požadavkové workshopy	116
Třídění získaných informací	118
Na co si dávat pozor	122
Hledání chybějících požadavků	123
Jak poznáte, že už je hotovo?	125
Další kroky	125

KAPITOLA 8**Jak správně pochopit uživatelské požadavky 127**

Případy užití	128
Případ užití versus scénář užití	129
Jak se případy užití hledají	132
Dokumentace případů užití	133
Případy užití a funkční požadavky	138
Výhody případů užití	140
Na co si dát pozor	141
Tabulky událostí a odpovědí	142
Další kroky	144

KAPITOLA 9**Hrajeme podle pravidel 145**

Pravidla podnikání	146
Fakta	147
Omezení	147
Aktivátory	148
Implikace	149
Výpočty	149
Dokumentace podnikatelských pravidel	150
Podnikatelská pravidla a požadavky	151
Další kroky	154

KAPITOLA 10**Dokumentace požadavků 155**

Specifikace požadavků	156
Značení požadavků	157
Co s chybějícími informacemi	158
Specifikace požadavků a uživatelská rozhraní	159
Šablona pro specifikaci požadavků	159
1. Úvod	161
2. Obecný popis	162
3. Funkce systému	163
4. Požadavky na vnější rozhraní	164
5. Další parametrické požadavky	166
6. Ostatní požadavky	167
Dodatek A: Slovníček	167
Dodatek B: Analytické modely	167
Dodatek C: Seznam úkolů	167

Tipy pro psaní požadavků	167
Příklady požadavků, předtím a poté	171
Další kroky	176

KAPITOLA 11

Jeden obrázek vydá za 1 024 slov 177

Modelování požadavků	178
Od zákazníka k analytickým modelům	179
Diagramy datových toků	180
ER diagramy	183
Přechodové diagramy	185
Mapy dialogů	188
Diagramy tříd	191
Rozhodovací tabulky a stromy	193
Poznámka na závěr	195
Další kroky	195

KAPITOLA 12

Kvalitativní parametry aneb Víc než funkce 197

Kvalitativní parametry	198
Definování kvalitativních požadavků	199
Parametry důležité pro uživatele	200
Parametry důležité pro vývojáře	205
Výkonnostní požadavky	206
Popis parametrických požadavků v jazyce Planguage	207
Kompromisy mezi parametry	208
Implementace parametrických požadavků	210
Další kroky	211

KAPITOLA 13

Snižování rizika prototypováním 213

Prototypování: Co a proč	214
Horizontální prototypy	214
Vertikální prototypy	215
Jednorázové prototypy	216
Evoluční prototypy	217
Papírové a elektronické prototypy	219
Hodnocení prototypů	220
Rizika prototypování	222
Klíčové faktory pro úspěch prototypování	223
Další kroky	224

KAPITOLA 14

Rozdělení priorit 225

Proč se požadavky dělí podle priority	226
Odstupňování priorit	228
Rozdělení priorit podle hodnoty, nákladů a rizika	229
Další kroky	233

KAPITOLA 15

Kontrola požadavků 235

Formální a neformální recenze	237
Inspekce požadavků	238
Účastníci	239
Role inspektorů	239
Vstupní podmínky inspekce	240
Fáze inspekce	241
Výstupní podmínky	243
Běžné chyby	244
Běžné překážky při kontrole požadavků	245
Velké dokumenty	245
Velké inspekční týmy	246
Zeměpisné rozdělení týmu	246
Testování požadavků	247
Jak definovat podmínky přijetí	252
Další kroky	253

KAPITOLA 16

Problémové typy projektů 255

Údržba starších systémů	255
Začněte sbírat informace	256
Zkoušejte nové techniky pro práci s požadavky	258
Všímejte si vztahů mezi požadavky	259
Aktualizujte dokumentaci	259
Krabicový software	260
Navrhněte případy užití	261
Berte ohled na podnikatelská pravidla	261
Popište kvalitativní požadavky	261
Outsourcované projekty	262
Experimentální projekty	264
Neformální specifikace uživatelských požadavků	265
Zákazník mezi vývojáři	265

Včasné a časté rozdělení priorit	266
Jednoduché řízení změn	266
Další kroky	266

KAPITOLA 17

Požadavky versus ostatní práce na projektu 267

Od požadavků k projektovým plánům	268
Požadavky a odhadování	270
Požadavky a termíny	271
Od požadavků k návrhu a kódu	272
Od požadavků k testům	275
Od požadavků k úspěchu	276
Další kroky	277

Část III

Správa požadavků

KAPITOLA 18

Principy a techniky pro správu požadavků 281

Směrná verze specifikace	283
Postupy pro správu požadavků	283
Verzování požadavků	284
Atributy požadavků	285
Sledování stavu požadavků	287
Jak sledovat náročnost správy požadavků	289
Další kroky	290

KAPITOLA 19

Když se přihodí změny 291

Jak se bránit proti narůstání požadavků	292
Politika řízení změn	294
Popis procesu pro řízení změn	295
Komise pro řízení změn	299
Složení komise	300
Zakládající dokument komise	301
Rozhodování	301
Oznamování stavu změn	301

Aktualizace závazků	301
Nástroje pro řízení změn	302
Sledování aktivity změn	302
Změny nejsou zadarmo: Analýza dopadu	304
Jak probíhá analýza dopadu	305
Šablona pro výsledky analýzy dopadu	309
Další kroky	310

KAPITOLA 20

Dohledatelnost požadavků **311**

Sledování požadavků	311
K čemu je dohledatelnost požadavků	314
Spojovací matice	315
Nástroje pro sledování požadavků	318
Zřizování odkazů	319
Dá se dohledatelnost požadavků zvládnout? Stojí za to?	320
Další kroky	321

KAPITOLA 21

Nástroje pro správu požadavků **323**

Výhody specializovaných programů	325
Funkce programů pro správu požadavků	327
Automatizace správy požadavků	329
Jak si vybrat vhodný nástroj	329
Jak změnit firemní kulturu	330
Jak zkrotit nástroje pro správu požadavků	331
Další kroky	332

Část IV

Řízení požadavků

KAPITOLA 22

Zlepšování stávajících procesů **335**

Požadavky a projektové procesy	336
Požadavky a účastníci projektu	337
Základy zlepšování softwarových procesů	339
Zlepšovací jednohubky	340
Cyklické zlepšování procesů	341
Analýza stávajících procesů	341
Plánování zlepšovacích prací	342

Návrh, zkušební provoz a nasazení nových procesů	344
Hodnocení výsledků	344
Procesní rekvizity	345
Rekvizity pro vývoj požadavků	347
Rekvizity pro správu požadavků	348
Plán zlepšování procesů	348
Další kroky	349

KAPITOLA 23

Softwarové požadavky a řízení rizik 351

Základy řízení softwarových rizik	352
Z čeho se skládá řízení rizik	353
Dokumentace rizik	354
Plánování rizik	356
Rizika spojená s požadavky	357
Sbírání požadavků	357
Analýza požadavků	359
Specifikace požadavků	359
Kontrola požadavků	359
Správa požadavků	360
Řízení rizik je váš kamarád	360
Další kroky	361

Epilog 363

Část V

Přílohy

PŘÍLOHA A

Dotazník pro posouzení vašich aktuálních procesů 367

PŘÍLOHA B

Vztah mezi požadavky a modely zlepšování procesů 373

Model SW–CMM	373
Model CMMI–SE/SW	375
Procesní oblast správy požadavků	376
Procesní oblast vývoje požadavků	377

PŘÍLOHA C

Hledání chyb v požadavcích **379**

Analýza příčin	379
Běžné příznaky problémů s požadavky	380
Běžné překážky řešení problémů	381

PŘÍLOHA D

Ukázková dokumentace požadavků **401**

Vize a rozsah projektu	402
Podnikatelské požadavky	402
Návrh řešení	403
Rozsah a omezení	404
Podnikatelský kontext	404
Případy užití	406
Objednání jídla	407
Srážení ze mzdy	409
Změna menu	411
Specifikace požadavků	412
Úvod	412
Obecný popis	413
Funkce systému	415
Požadavky na vnější rozhraní	418
Další parametrické požadavky	419
Dodatek A: Datový slovník a datový model	420
Dodatek B: Analytické modely	423
Podnikatelská pravidla	424

Slovníček **425****Rejstřík** **435**

O autorovi

Karl E. Wiegers je hlavním poradcem ve firmě Process Impact, která se zabývá výukou a poradenstvím v oblasti softwarových procesů a sídlí v oregonském Portlandu. Školil a radil už desítkám firem po celém světě. Dříve pracoval 18 let ve firmě Eastman Kodak, nejprve jako fotografický výzkumník, pak jako softwarový vývojář, softwarový manažer a nakonec jako vedoucí pro softwarové procesy a zlepšování kvality. Má bakalářský titul v oboru chemie ze Státní univerzity v Boise a magisterský a doktorandský titul v oboru organické chemie z Illinoiské univerzity. Je členem IEEE, IEEE Computer Society a ACM.

Karl je autorem knih *Peer Reviews in Software: A Practical Guide* (Addison-Wesley 2002) a *Creating a Software Engineering Culture* (Dorset House 1996), z nichž ta druhá mu vynesla prestižní *Productivity Award* časopisu *Software Development*. Napsal asi 160 článků o mnoha stránkách výpočetní techniky, chemie a vojenské historie. Mimo jiné přispívá do časopisu *Software Development* a je členem redakce časopisu *IEEE Software*.

Když zrovna není před obrazovkou nebo za katedrou, užívá si hru na své kytary Gibson Les Paul, Fender Stratocaster a Guild D40, jezdí na své motorce Suzuki VX800, studuje historii válčení, vaří a popíjí víno se svou ženou Chris Zambitovou. Karla můžete kontaktovat na stránkách www.processimpact.com.

Úvodem

Softwarovému průmyslu už je nějakých padesát let, ale hodně softwarových firem přesto stále ještě zápasí se sbíráním, dokumentací a správou požadavků kladených na jejich vlastní výrobky. Nedostatek informací od uživatelů, neúplné požadavky a jejich neustálé změny jsou hlavními důvody, kvůli kterým tolik projektů z oblasti informačních technologií nezvládne dodat slibovanou funkcionalitu v daném termínu za daný rozpočet.¹ Řada softwarových vývojářů se při získávání informací od uživatelů necítí pohodlně a nebo v něm nemá dostatek praxe. Zákazníci zase spolupráci na definici požadavků odmítají z nedostatku trpělivosti a nebo nechají požadavky sepsat nevhodné lidi. Účastníci projektu se mnohdy ani nedohodnou na tom, co vlastně takový požadavek je. Jak poznamenal jeden spisovatel: „Než by se programátoři pokusili o dešifrování zákaznickových požadavků, raději by luštili slova klasické písně Louie Louie od The Kingsmen.“²

Z praxe

Při vývoji softwaru je komunikace potřeba přinejmenším stejně jako programování, ale my přesto běžně dáváme přednost programování. Tato kniha nabízí desítky nástrojů, které komunikaci usnadní a pomohou softwarovým vývojářům, manažerům, obchodníkům i zákazníkům použít efektivní postupy práce s požadavky. Ve druhém vydání (originálního titulu) přibýly kapitoly o roli analytika požadavků, důležitosti podnikatelských pravidel a různých způsobech, kterými se dá práce s požadavky aplikovat na udržování projektů, krabicový software, outsourcované projekty a projekty s neurčitými a měnícími se požadavky. V textu najdete bezpočet historek (všech skutečných!), které dokreslují běžné zkušenosti týkající se řízení požadavků.

Všechny uváděné postupy se v řízení požadavků řadí do hlavního proudu, nejde o žádné exotické nové techniky nebo složité metodologie, které by slibovaly vyřešit všechny vaše problémy s požadavky. Od roku 1999, kdy jsem napsal první vydání této knihy, jsem odučil přes 100 seminářů o softwarových požadavcích pro klienty ze soukromých i státních organizací všech druhů. Poznal jsem, že se tyto postupy vztahují prakticky na libovolný projekt, včetně projektů vedených inkrementálním stylem – na nové systémy i na udržování těch starých, na malé projekty i na ty velké. Neomezují se dokonce vůbec na softwarové projekty, dají se vztáhnout i na hardware a výrobu jakýchkoliv jiných systémů. Stejně jako u jakékoliv jiné softwarové techniky platí, že pokud je máte využít v nejlepší možné míře, musíte získat nějaké zkušenosti a spolehout se na svůj zdravý rozum.

¹ *The CHAOS Report*, Standish Group International, 1995.

² *Risque Requirements*, Gary Peterson, 2002, CrossTalk 15(4):31.

Výhody, které vám tato kniha nabízí

Zlepšení práce s požadavky vám pravděpodobně přinese větší výhody, než zlepšení jakéhokoliv jiného softwarového procesu. Budeme se soustředit na praktické, ověřené postupy, které vám pomohou:

- Zlepšit kvalitu požadavků na váš systém v samotném začátku vývojového cyklu, čímž zmenšíte objem předělávané práce a zvýšíte svou produktivitu.
- Zvládnout tiché narůstání rozsahu projektu i změny požadavků a splnit tak stanovené termíny.
- Dosáhnout vyšší spokojenosti zákazníků.
- Snížit náklady na údržbu a podporu vašeho systému.

Chtěl bych vám pomoci se zlepšováním procesů pro sbírání a analýzu požadavků, psaní a revize specifikací těchto požadavků a jejich správu v průběhu celého vývojového cyklu. Doufám, že si o lepších postupech nebudete jen číst, a skutečně je nasadíte v praxi. Nastudovat si něco o nových postupech je hračka – mnohem složitější je změnit způsob, kterým lidé pracují.

Pro koho je tato kniha určena

V této knize najde užitečné informace každý, kdo potřebuje popsat nebo pochopit požadavky na softwarový systém. Hlavní cílovou skupinou jsou členové vývojářského týmu, kteří hrají úlohu analytika požadavků, a všichni další, již se k této roli čas od času dostanou. Další cílovou skupinou jsou návrháři, programátoři, testeři a ostatní členové týmu, kteří musí pochopit a splnit požadavky uživatelů. Popisované postupy se budou hodit i obchodníkům a produktovým manažerům, kteří jsou za příslušný systém zodpovědní a mají navrhnout funkce a vlastnosti nezbytné pro jeho úspěch na trhu. Projektoví manažeři, kteří musí ve stanovených termínech odevzdávat svěřené projekty, se zde naučí zvládat činnosti spojené s požadavky na projekt a vypořádat se s jejich změnami. A konečně třetím typem publika jsou zákazníci, kteří chtějí umět přesně definovat své požadavky na funkce a kvalitu systému. Tato kniha jim pomůže pochopit důležitost procesu zpracování požadavků a důležitost role, kterou v něm hrají.

Co v knize najdete

Tato kniha je rozdělena do čtyř částí. Část první, nazvaná *Softwarové požadavky: Co, proč a kdo*, začíná několika definicemi a popisem několika vlastností dobře napsaných požadavků. Pokud patříte do technického týmu, doufám, že si druhou kapitolu věnovanou vztahu mezi vývojářem a zákazníkem nenecháte pro sebe a podělíte se o ni se svými klíčovými zákazníky. Kapitola třetí popisuje několik desítek kvalitních průmyslových postupů pro vývoj požadavků, jejich správu a práci s požadavky obecně. Předmětem čtvrté kapitoly jsou úkoly, které čekají na analytika požadavků.

Druhá část knihy, *Vývoj softwarových požadavků*, začíná technikami pro definování podnikatelských požadavků projektu. Další kapitoly se věnují hledání vhodných zákaznickových zástupců, získávání jejich požadavků a popisování různých případů užití, podnikatelských

pravidel, funkčních požadavků a kvalitativních parametrů. Kapitola 11 popisuje několik analytických modelů, díky kterým se na požadavky můžete podívat z několika různých pohledů, a kapitola 13 se zabývá snižováním rizika pomocí softwarových prototypů. Zbývající kapitoly druhé části představují nástroje pro rozdělení priorit a kontrolu požadavků. Na závěr druhé části se dozvíte o problémech, které pro analýzu požadavků představují některé konkrétní situace, a o vlivech požadavků na další práce na projektu.

Předmětem třetí části jsou principy a postupy pro správu požadavků, s důrazem na řešení změn požadavků. Kapitola 20 popisuje dohledatelnost požadavků, tedy spojení požadavků s jejich autory a jednotlivými částmi hotového systému. Tuto část uzavře popis komerčních nástrojů, které mohou vaši správu požadavků vylepšit.

Poslední část knihy se zabývá zaváděním procesů pro práci s požadavky do praxe. Kapitola 22 vám pomůže prosadit nové postupy pro řízení požadavků do vývojového procesu vašeho týmu, běžná rizika spojená s požadavky popisuje kapitola 23 a příloha A vám pomůže posoudit, které z vašich postupů jsou zralé na zlepšení. V dalších přílohách pak najdete průvodce řešením běžných problémů v požadavcích a příklady několika specifikací požadavků.

Případové studie

K dokreslení postupů popisovaných v této knize jsme přidali několik ukázek z případových studií skutečných projektů, především středně velkého informačního Systému pro evidenci chemikálií. (A nebojte – budete jim rozumět i bez jakýchkoliv znalostí chemie.) Na mnoha místech najdete také příklady rozhovorů mezi jednotlivými účastníky projektů z těchto případových studií. Podle našeho názoru se vám budou hodit bez ohledu na to, jaký typ softwaru váš tým dělá.

Od slov k činům

Dát dohromady dost energie k uplatnění nových znalostí v praxi je těžké. Staré známé, byť třeba nepříliš efektivní, postupy představují velké pokušení. S nasazováním nových procesů vám na konci každé kapitoly pomůže odstavec *Další kroky*, ve kterém se dozvíte, jak obsah příslušné kapitoly použít v praxi. Komentované šablony pro dokumentaci požadavků, seznamy úkolů, tabulku pro rozdělení priorit požadavků, proces řízení změn a mnoho dalších pomůcek najdete na mé webové stránce www.processimpact.com. S těmito materiály se můžete do vylepšování pustit hned. Vylepšujte pomalu, ale začněte už dnes.

Některým účastníkům projektu se do nových postupů nebude chtít. Na některé lidi rozumné důvody prostě nezabírají, a s takovými lidmi vám žádá z popisovaných technik nepomůže. Snažte se pomocí materiálů z této knihy vzdělávat své kolegy, zákazníky i manažery. Upozorňujte je na problémy, které v souvislosti s požadavky vznikly u předchozích projektů, a proberte potenciální výhody nových přístupů.

Při zlepšování postupů pro řízení požadavků nemusíte čekat na nový projekt – šestnáctá kapitola se zabývá tím, jak většinu postupů z této knihy aplikovat i na údržbu starších systémů. Postupné zlepšování postupů je celkem bezpečný proces, kterým můžete zvýšit kvalitu

práce na stávajícím projektu a zároveň se připravit na použití nových postupů u svého dalšího projektu.

Cílem řízení požadavků je dostatečně kvalitní definice požadavků, podle které se můžete do návrhu a implementace pustit za přijatelné úrovně rizika. Pokud chcete minimalizovat riziko předělávání hotových částí projektu, vývoje špatného systému a překročení stanovených termínů, musíte nad požadavky strávit dostatečně dlouhou dobu. Tato kniha vám dá nástroje, díky kterým se sejdou ti správní lidé a dojdou ke správným požadavkům na správný produkt.

Poznámka redakce českého vydání

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

Computer Press
redakce počítačové literatury
Holandská 8
639 00 Brno

nebo

knihy@cpress.cz.

Další informace a případné opravy českého vydání knihy najdete na internetové adrese <http://knihy.cpress.cz/k1567>. Prostřednictvím uvedené adresy můžete též naší redakci zaslat komentář nebo dotaz týkající se knihy. Na vaše reakce se srdečně těšíme

Poděkování

Na přečtení rukopisu a nepočítaně dobrých rad k jeho zlepšení si našla čas řada lidí, mají můj hluboký vděk. První vydání četli Steve Adolph, Nikki Bianco, Bill Dagg, Dale Emery, Chris Fahlbusch, Geoff Flamank, Lynda Flemingová, Kathy Getzová, Jim Hart, Tammy Hoganson, Mike Malec, Deependra Moitraová, Mike Rebatzke, Phil Recchio, Kathy Rhodeová, Johanna Rothmanová, Joyce Statzová, Doris Sturzenbergerová, Prakash Upadhyayula a Scott Whitmire. K tomuto druhému vydání mi cennými poznámkami přispěli Steve Adolph, Ross Collard, Richard Dalton, Chris Fahlbusch, Lynda Flemingová, Ellen Gottesdienerová, Linda Lewisová, Jeff Pink, Erik Simmons, David Standerford, Dennis Stephenson, Scott Whitmire, Rebecca Wirfs-Brocková a Trish Zwirnbaumová.

Za první vydání si mé poděkování zaslouží mnoho lidí z nakladatelství Microsoft Press, mimo jiné redaktoři Ben Ryan, John Pierce, Mary Kalbach Barnardová a Michelle Goodmanová, výtvarník Rob Nance a sazečka Paula Gorelicková. Za druhé vydání bych chtěl poděkovat redaktorce Danielle Birdové, redaktorům Thomasovi Pohlmannovi, Devonovi Musgraveovi, redaktorce Sandi Resnickové, výtvarníkovi Michaelovi Kloepferovi a sazečce Gině Cassillové. Velice užitečné byly také stovky příspěvků a komentářů, které jsem za několik posledních let nasbíral na svých seminářích o softwarových požadavcích. Kdybyste se se mnou chtěli podělit o své zkušenosti, napište mi, prosím, na adresu kwiegers@acm.org.

Můj největší dík patří Chris Zambitové, té nejvtipnější, nejtrpělivější ženě a podpoře, jakou by si kdy autor mohl přát.

Část I

Podstata a význam požadavků na software

KAPITOLA 1

Nejdůležitější požadavek na software

„Haló, mluvím s Filipem? Tady Marie z osobního. Máme problém s tím zaměstnaneckým systémem, který jste pro nás programoval. Jedna slečna si zrovna změnila jméno na Jiskru Jasnou a my tu změnu jména nemůžeme dostat do systému. Můžete nám pomoci?“

„Čili ona se provdala za nějakého pana Jasného?“

„Ne, neprovdala se, jen si změnila jméno,“ odpověděla Marie. „V tom je právě ten problém. Ono to vypadá, že můžeme jméno změnit jen při změně stavu.“

„No to ano, protože mě nikdy nenapadlo, že by si někdo jen tak měnil jméno. Nepamatuju se, že byste mi o téhle možnosti říkala, když jsme spolu o systému mluvili. Proto se do dialogu pro změnu jména dá dostat jen z dialogu pro změnu stavu,“ vysvětlil Filip.

„Myslela jsem, že víte, že si kdokoliv může změnit jméno kdykoliv chce,“ namítla Marie. „Musíme to vyřešit do pátku, jinak si Jiskra nemůže vyzvednout výplatu. Stihnete to spravit?“

„Není co spravovat, to není žádná chyba!“ opáčil Filip. „Vůbec jsem netušil, že takovou funkci potřebujete. Zrovna mám rozdělaný nový systém na analýzu výkonu a myslím, že tu mám i nějaké další požadavky na databázi zaměstnanců.“ [šustění papíru] „Jo, tady je další. Asi bych to stihl do konce měsíce, ale do konce týdne určitě ne. Je mi líto; příště mi takové věci musíte říkat předem a dát mi je černé na bílém.“

„A co mám říct slečně Jiskře?“ dožadovala se Marie. „Bude pěkně namíchnutá, když si nebude moct vyzvednout výplatu.“

„Podívejte, Marie, ale tohle přece není moje chyba,“ protestoval Filip. „Kdybyste mi už na začátku řekla, že chcete mít možnost kdykoliv změnit něčí jméno, k tomuhle by vůbec nedošlo. Nemůžete mi vyčítat, že vám nevidím do hlavy.“

Marie se zlostně a odevzdaně utrhla: „Tak tohle je přesně to, kvůli čemu nesnáším počítače. Buďte, prosím, tak laskav a zavolejte mi hned, jak to bude spravené, ano?“

Pokud jste se někdy jako zákazník zúčastnili podobného rozhovoru, máte představu, jak frustrující je používat software¹, který vám nedovolí udělat nějakou základní věc. A určitě byste nechtěli být vydáni na milost vývojáři, který se k vašemu zásadnímu požadavku na změnu *možná časem dostane*. Vývojáři vědí, jak frustrující je dozvědět se po dokončení systému o nějaké funkci, kterou od něj uživatelé očekávali. Nepříjemná je i nutnost přerušit práci na aktuálním projektu kvůli požadavku na změnu systému, který dělá přesně to, co měl.

Hodně problémů se softwarem plyne z nedostatků při sběru, dokumentaci, schvalování a úpravách požadavků na výsledný systém. Mezi problémové oblasti může – stejně jako u Filipa a Marie – patřit neformální získávání informací, mlčky předpokládaná funkcionality, chybné nebo zamlčené předpoklady, nedostatečně definované požadavky a neformální, nahodilé, změny.

Většinu lidí by nenapadlo postavit nový dům za šest milionů, aniž by své potřeby a požadavky důkladně probrali s projektantem a všechny podrobnosti postupně upřesnili. Jasně chápou, že jakékoliv změny domu s sebou nesou náklady – nelíbí se jim to, ale rozumí tomu. Naproti tomu u vývoje softwaru se podobné problémy lehkovážně přehlíží. Chyby způsobené při analýze požadavků mohou za 40–60 procent všech chyb softwarových projektů (Davis 1993, Leffingwell 1997). Při velkém průzkumu evropského softwarového trhu se přišlo na to, že dva nejčastěji zmiňované problémy jsou popis a správa zákaznických požadavků (ESPITI 1995). Řada organizací pro tyto základní procesy přesto používá neefektivní metody. Obvyklým výsledkem pak je rozdíl v očekávání: rozdíl mezi tím, co se vývojáři chystají napsat, a tím, co zákazník skutečně potřebuje.

Nikde se zájmy všech účastníků softwarového projektu nesetkávají tak, jako ve fázi analýzy požadavků. Mezi účastníky patří:

- Zákazníci, kteří projekt financují nebo chtějí dostat systém, jenž pokryje jejich podnikatelské potřeby.
- Uživatelé, kteří se systémem přímo či nepřímo pracují (podmnožina zákazníků).
- Analytici požadavků, kteří sepisují požadavky na systém a tlumočí je vývojářům.
- Vývojáři, kteří systém navrhují, implementují a udržují.
- Testeři, kteří zjišťují, jestli se systém skutečně chová, jak má.
- Dokumentátoři, kteří píšou uživatelské příručky, školicí materiály a nápovědu.
- Vedoucí projektu, kteří projekt plánují a vedou vývojářský tým k úspěšnému odevzdání softwaru.
- Právníci, kteří zodpovídají za to, že systém odpovídá všem příslušným zákonům a nařízením.
- Lidé z výroby, pokud je software součástí nějakého výrobku.
- Lidé z prodejního oddělení, marketingového oddělení, technické podpory a všichni další, kteří budou se systémem nebo jeho uživateli muset pracovat.

¹ Výrazy *software*, *aplikace*, *produkt* a *systém* budu v knize volně zaměňovat. Probírané principy a postupy se vztahují na libovolné programy a systémy se softwarem, které kdy vytvoříte.

Když se tento průnik pojme dobře, dostanete zajímavý systém, nadšené zákazníky a spokojené vývojáře. Když se pojme špatně, dostanete zdroj nedorozumění, frustrací a konfliktů, které snižují kvalitu systému a jeho hodnotu pro zákazníka. Jelikož jsou požadavky základem pro vývoj softwaru a všechny úkoly spojené s řízením projektu, účastníci se musí zavázat, že s nimi budou pracovat efektivně.

Jenže vývoj a řízení požadavků je problém. Žádné zkratky ani magická řešení neexistují. Hodně organizací se naštěstí potýká s podobnými problémy, takže můžeme hledat společné postupy, které se budou hodit ve větším počtu situací. Desítky takových postupů ukazuje právě tato kniha. Jsou popisované z pohledu návrhu nového systému, ale většina z nich se dá stejně dobře použít pro udržování starších projektů a výběr krabicového softwaru. (Viz kapitolu 16, *Problémové typy projektů*.) Stejně tak se tyto techniky netýkají pouze projektů, které používají vývojový cyklus vodopád – i týmy, které své projekty navrhují přírůstkově, musí jasně vědět, co se při každé vývojové iteraci děje.

Tato kapitola vám pomůže:

- Pochopit některé klíčové pojmy, jež se v oboru softwarových požadavků používají.
- Rozlišit vývoj požadavků od jejich správy.
- Uvědomit si některé problémy, jež bývají s požadavky spojené.
- Dozvědět se několik vlastností dobře formulovaných požadavků.

Jak jste na tom vy?

Možná byste chtěli v rychlosti posoudit postupy, které se pro práci s požadavky používají ve vaší organizaci. V takovém případě se podívejte, kolik z následujících bodů se týká vašeho posledního projektu. Pokud zaškrtnete více než tři nebo čtyři okénka, takhle kniha je pro vás.

- Cíl projektu a jeho rozsah nejsou nikdy jasně definované.
- Zákazníci mají moc práce na to, aby s analytiky nebo vývojáři trávili čas nad požadavky.
- Zástupci uživatelů (například manažeři nebo marketingoví pracovníci) tvrdí, že mluví za uživatele, ale požadavky uživatelů nepředávají dostatečně přesně.
- Ve vaší organizaci drží požadavky v hlavě „odborníci“, požadavky se nikdy nedostanou na papír.
- Zákazníci tvrdí, že jsou pro ně všechny požadavky kritické, takže je nerozdělují podle priorit.
- Vývojáři při implementaci narážejí na nejednoznačné požadavky a chybějící informace, takže si musí domýšlet.
- Komunikace mezi vývojáři a zákazníky se soustředí na obrázky uživatelského rozhraní a nikoliv na to, co uživatelé se softwarem potřebují dělat.
- Vaši zákazníci schválí seznam požadavků a pak jej neustále mění.
- Rozsah projektu se s dalšími změnami požadavků rozšiřuje, ale nedostanete další zdroje a nedojde k odstranění žádné funkcionality, takže vám utíkají termíny.

- Vyžádané změny v požadavcích se ztrácejí a vy ani vaši zákazníci nemáte přehled o stavu jednotlivých změn.
- Zákazníci si vyžádají nějakou funkcionalitu a vývojáři ji naprogramují, ale nikdo ji nikdy nepoužije.
- Systém přesně splní specifikaci, ale zákazník není spokojený.

Definice softwarových požadavků

Jedním z problémů softwarového průmyslu je nedostatek společných definic. Různí pozorovatelé mohou tutéž větu popsat jako uživatelský požadavek, softwarový požadavek, funkční požadavek, systémový požadavek, technický požadavek nebo podnikatelský požadavek. Zákazníková definice požadavků zní vývojářům nejspíš jako vysoká abstrakce, popis principů; a naopak vývojářova představa o požadavcích zní uživateli pravděpodobně jako něco z oboru návrhu uživatelských rozhraní. Z tohoto bohatého výběru definic plynou matoucí a frustrující problémy v komunikaci.

Z praxe

Klíčová myšlenka zní, že požadavky musíte dokumentovat. Byl jsem u jednoho projektu, na kterém se střídali vývojáři. Hlavnímu zákazníkovi už bylo do pláče, když k němu přišel nový analytik požadavků a řekl, že si musí promluvit o jeho požadavcích. „Své požadavky už jsem dal vašim předchůdcům, teď mi konečně napište ten systém!“ Ve skutečnosti se nikdo nenamáhal požadavky zapsat, takže každý nový analytik musel začínat úplně odznovu. Pokud říkáte, že požadavky dokumentované máte, a myslíte tím hromadu e-mailů, záznamů z hlasové schránky, poznámkových papírků, zápisů ze schůzek a několik matně zapamatovaných rozhovorů na chodbě, žijete v klamu.

Několik výkladů slova požadavek

Poradce Brian Lawrence navrhuje, že *požadavek* je „cokoliv, co ovlivňuje rozhodování při návrhu“ (Lawrence 1997). Do této kategorie se vejde hodně druhů informací. Slovníček softwarové terminologie (*IEEE Standard Glossary of Software Engineering Terminology*) z roku 1990 definuje požadavek jako:

1. Podmínku nebo funkci, kterou uživatel potřebuje pro řešení problému nebo dosažení nějakého cíle.
2. Podmínku nebo funkci, kterou musí systém nebo jeho část splňovat, aby vyhověl smlouvě, standardu, specifikaci nebo jinému dokumentu, jenž se na něj formálně vztahuje.
3. Dokumentovanou podobu některého z předchozích dvou bodů.

Tato definice zahrnuje uživatelský pohled na požadavky (vnější chování systému) i vývojářův pohled (některé vnitřní parametry). Výraz *uživatel* by se měl nahradit obecnějším *účastník projektu* nebo prostě *účastník*, protože ne všichni účastníci jsou uživatelé. Já si požadavek představuji jako vlastnost, již systém musí mít, aby měl hodnotu pro některého z účastníků.

Rozmanitost různých druhů požadavků uznává i následující definice (Sommerville a Sawyer, 1997):

Požadavky jsou (...) popis toho, co všechno by se mělo implementovat. Popisují žádané chování systému a jeho vlastnosti a mohou představovat nějaká omezení procesu vývoje systému.

Je zjevné, že žádná univerzální definice požadavků není. Pokud si máme usnadnit komunikaci, musíme se shodnout na jednotné skupině přídavných jmen, kterými se přetížené slovo požadavek upřesní, a musíme se naučit ocenit hodnotu obecně srozumitelného zápisu těchto požadavků.



Upozornění

Nepředpokládejte, že všichni účastníci vašeho projektu sdílí jednu společnou představu o tom, co jsou to požadavky. Ujasněte si definici předem, abyste všichni mluvili o tomtéž.

Typy požadavků

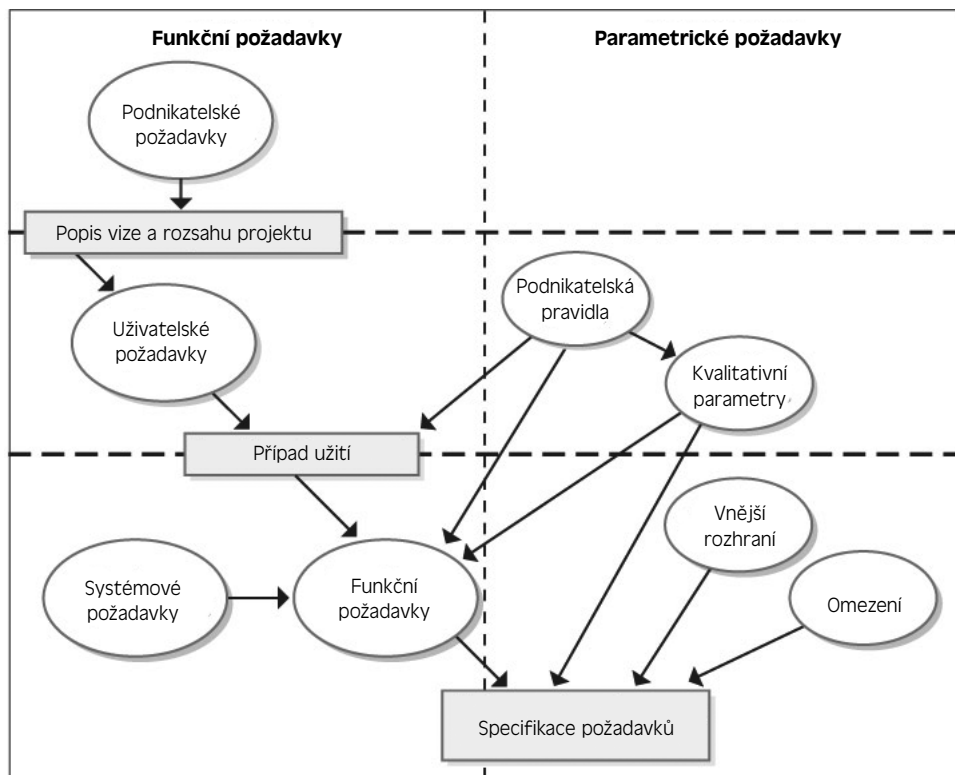
V této části najdete definice, které budu používat pro některé běžné výrazy z oblasti softwarových požadavků. Požadavky na software se dají rozdělit do tří různých skupin: na podnikatelské, uživatelské a funkční. Kromě nich má každý systém ještě zásobu takzvaných parametrických požadavků, které vyplývají nikoliv z jeho funkce, ale jeho prostředí. Vztahy mezi těmito typy požadavků ukazuje model na obrázku 1.1. Jak už to u modelů bývá, nepokrývá úplně všechno, ale přesto poskytuje užitečnou organizační oporu. Elipsy představují různé typy požadavků a obdélníky různé způsoby jejich uložení, například dokumenty, diagramy nebo databáze.



Další informace

Hodně příkladů různých typů požadavků obsahuje sedmá kapitola, *Jak nepřeslechnout zákazníkův hlas*.

Podnikatelské požadavky formulují na nejvyšší úrovni cíle organizace nebo zákazníka, který si o systém řekl. Většinou pochází od hlavního investora, nabývajícího zákazníka, vedoucího uživatelů, marketingového oddělení nebo produktového vizionáře. Podnikatelské požadavky říkají, proč vlastně organizace systém chce – označují cíle, kterých by organizace prostřednictvím systému ráda dosáhla. Já podnikatelské požadavky většinou zapisuji ve formě samostatného dokumentu – takzvané *projektové charty*, která popisuje vize a rozsah projektu (psaní tohoto dokumentu je předmětem páté kapitoly). Jasná definice rozsahu projektu je prvním krokem ke zvládnutí běžného problému s tichým narůstáním rozsahu.



Obrázek 1.1. Vztahy mezi různými typy požadavků.

Uživatelé požadavky popisují cíle uživatelů a úkoly, které musí být uživatelé schopni se systémem provést. Mezi užitečné způsoby zápisu uživatelských požadavků patří případy užití, scénáře a tabulky popisující reakce na různé události. Uživatelé požadavky tedy popisují, co bude uživatel se systémem schopen dělat. Příkladem případu užití je třeba rezervace na webové stránce letecké společnosti, hotelu nebo půjčovny aut.



Další informace

Uživatelé požadavky se zabývá kapitola osmá, *Jak správně pochopit uživatelské požadavky*.

Funkční požadavky popisují softwarovou funkcionalitu, kterou musí vývojáři do systému dostat, aby uživatelé mohli splnit své úkoly, a tím i podnikatelské požadavky. Funkčním požadavkům se někdy též říká *behaviorální požadavky*, neboli požadavky na chování. Právě mezi ně patří ony klasické věty typu *měl by*: „Potvrzení o rezervaci by systém měl odeslat uživateli.“ Jak uvidíte v desáté kapitole nazvané *Dokumentace požadavků*, funkční požadavky určují, co přesně musí vývojáři naprogramovat.

Systémové požadavky jsou celkové požadavky na výrobek složený z většího počtu podsystémů, tedy *systém* ve smyslu IEEE 1998c. Systém může být plně softwarový, nebo může softwarové

podsystemy kombinovat s hardwarovými. Součástí systému jsou i lidé, takže některé systémové funkce mohou být svěřené lidem.

Z praxe

Můj tým například jednou psal software pro ovládání nějakého laboratorního přístroje, který automatizoval pracné odměřování přesného množství chemikálií do skupiny kádinek. Požadavky na systém jako celek nám posloužily pro odvození funkčních požadavků na software, tedy posílání signálů hardwaru, aby přesunul trysku na chemikálie, přečetl stav polohových senzorů a zapnul nebo vypnul pumpu.

Mezi *podnikatelská pravidla* patří firemní předpisy, státní nařízení, průmyslové standardy, systémy účetnictví nebo výpočetní algoritmy. V deváté kapitole nazvané *Jak hrát podle pravidel* se dozvíte, že podnikatelská pravidla sama o sobě nepatří mezi požadavky na software, protože existují mimo hranice libovolného konkrétního softwarového systému. Často ale omezují počet lidí, již připadají v úvahu pro některý způsob použití systému, a nebo předepisují funkce, bez nichž by systém nesplnil patřičná pravidla. Z podnikatelských pravidel někdy plynou kvalitativní parametry, které vedou ke konkrétní funkcionalitě. To znamená, že některé funkce systému se dají zpětně dohledat až ke konkrétním podnikatelským požadavkům.

Všechny tyto funkční požadavky jsou popsány v dokumentu jménem *software requirements specification* (SRS), neboli *specifikace požadavků*, která do všech potřebných podrobností popisuje očekávané chování výsledného systému. O specifikaci požadavků budu mluvit jako o dokumentu, ale může to být i databáze, tabulka, informace uložené v komerčním programu pro správu požadavků (viz kapitolu 21, *Nástroje pro správu požadavků*) a u malých projektů dokonce třeba jen hromádka kartotéčních lístků. Specifikace požadavků se používá při vývoji, testování, řízení kvality, vedení projektu a dalších příbuzných pracích na projektu.

Kromě funkčních požadavků specifikace obsahuje i požadavky na parametry systému. Mezi ty patří například požadavky na výkonnost systému a kvalitativní parametry. *Kvalitativní parametry* rozšiřují popis funkcí systému v různých směrech, na kterých záleží uživatelům nebo vývojářům – popisují například použitelnost, přenositelnost, integritu, rychlost a odolnost systému. *Omezení* vývojářům při návrhu a vývoji systému zakazují použití některých cest.

Často se mluví o *features*, neboli funkcích systému. *Funkce* je skupina logicky souvisejících funkčních požadavků, které představují nějaký nástroj pro uživatele a umožňují splnění některého z podnikatelských požadavků. Ve světě komerčních programů je funkce skupina požadavků, která zákazníkovi pomáhá rozhodnout ve prospěch koupě; jeden bod v popisu výrobku. Seznam funkcí, které zákazník od výrobku požaduje, je něco jiného než úplný popis jeho požadavků na funkčnost. Příkladem funkcí jsou záložky webového prohlížeče, kontrola překlepů, nahrávání maker, elektrické stahování oken v autě, online aktualizace daňových kódů, ukládání často volaných telefonních čísel nebo automatické aktualizace virových databází. Jedna taková funkce se může týkat většího počtu případů užití a každý případ užití vyžaduje implementaci mnoha různých funkčních požadavků.

Abyste měli lepší představu o různých typech požadavků, o kterých jsme právě mluvili, představte si textový procesor. Jeden z podnikatelských požadavků by mohl znít například takto:

„Systém uživatelům umožní snadnou opravu překlepů v dokumentech.“ Obal krabice bude hlásat, že program obsahuje funkci jménem kontrola překlepů, která tento podnikatelský požadavek splňuje. Mezi odpovídající uživatelské požadavky mohou patřit případy užití jako „Hledání překlepů“ nebo „Přidání slova do systémového slovníku“. Kontrola překlepů má řadu samostatných funkčních požadavků, které se týkají operací jako vyhledávání a zvýraznění špatně napsaného slova, zobrazení dialogu s doporučovými náhradami nebo nahrazení překlepu v celému dokumentu. Kvalitativní parametr jménem *použitelnost* by pak řekl, co přesně mělo v podnikatelském požadavku znamenat slovo „snadno“.

Podnikatelské požadavky určují manažeři nebo marketingoví pracovníci s ohledem na to, aby jejich firma pracovala efektivněji (což se týká informačních systémů) nebo dokázala lépe konkurovat jiným firmám na trhu (to se týká komerčního softwaru). S podnikatelskými požadavky musí být v souladu všechny uživatelské požadavky, kterými analytik říká, jak uživatel v systému splní své úkoly. Funkční a parametrické požadavky pak používají vývojáři, kteří podle nich implementují požadovanou funkcionalitu a v rámci заданých omezení dosáhnou předepsaných kvalitativních a výkonnostních měřítek.

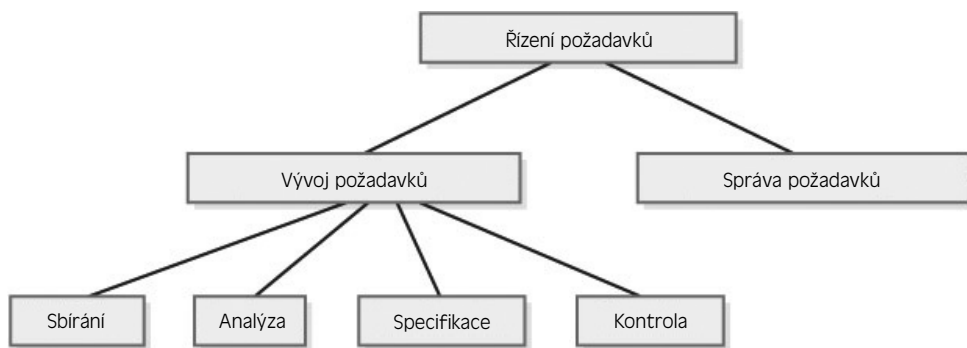
Na modelu z obrázku 1.1 jdou sice požadavky odshora dolů, ale v praxi byste měli očekávat několik iterací celého procesu a postupné upřesňování podnikatelských, uživatelských i funkčních požadavků. Kdykoliv někdo navrhne novou funkci, nový případ užití nebo nový funkční požadavek, analytik se musí zeptat: „Patří tohle ještě do vymezeného rozsahu systému?“ Pokud je odpověď kladná, požadavek do specifikace patří. Pokud je odpověď záporná, požadavek do specifikace nepatří. Pokud odpověď zní „ne, ale mělo by“, autor podnikatelských požadavků nebo hlavní investor se musí rozhodnout, jestli se podle nového požadavku upraví rozsah projektu. To je podstatné rozhodnutí, které má vliv na termín dokončení a rozpočet projektu.

Co mezi požadavky nepatří

Do specifikace požadavků nepatří podrobnosti o návrhu nebo implementaci (s výjimkou známých omezení), informace o plánování projektu ani informace o testování (Leffingwell a Widrig 2000). Tyto údaje od požadavků oddělte, aby se specifikace požadavků mohla soustředit na to, co vlastně má vývojářský tým napsat. Projekty mívají i další požadavky, například požadavky na vývojové prostředí, termín, rozpočet, školení uživatelů nebo požadavky na vydání systému a jeho přesun do fáze podpory. To jsou ale požadavky na samotný *projekt*, nikoliv *produkt*, a do tématu této knihy už nepatří.

Vývoj a správa požadavků

Terminologické zmatky týkající se požadavků sahají až k názvu celé disciplíny. Někteří autoři mluví o *řízení požadavků* neboli *requirements engineering* (Sommerville a Kotonya 1998), jiní o *správě požadavků* čili *requirements management* (Leffingwell a Widrig 2000). Mně přišlo praktické rozdělit práci se softwarovými požadavky na *vývoj požadavků* (kterým se zabývá druhá část této knihy) a *správu požadavků* (adresovanou v části třetí), viz obrázek 1.2.



Obrázek 1.2. Jednotlivé disciplíny práce s požadavky.

Vývoj požadavků

Vývoj požadavků můžeme dále rozdělit na jejich *sbírání*, *analýzu*, *specifikaci* a *kontrolu* (Abran a Moore 2001). Tyto disciplíny pokrývají všechny úkoly spojené se získáváním, vyhodnocováním a dokumentací požadavků na software nebo systém, který software obsahuje jako jednu ze svých částí. Mezi zmíněné úkoly patří například:

- Identifikace tříd uživatelů, kteří budou systém používat.
- Získávání požadavků od zástupců jednotlivých tříd.
- Pochopení jednotlivých uživatelských úkolů a podnikatelských cílů, které se tyto úkoly snaží splnit.
- Analýza informací od uživatelů a odlišení jejich cílů od funkčních požadavků, parametrických požadavků, podnikatelských požadavků, navrhovaných řešení a nadbytečných informací.
- Rozdělení části požadavků mezi softwarové moduly dané architekturou systému.
- Seřazení kvalitativních parametrů podle důležitosti.
- Vyjednání implementačních priorit.
- Zpracování nasbíraných uživatelských potřeb do podoby modelů a psané specifikace požadavků.
- Kontrola dokumentovaných požadavků, aby byla jistota, že uživatelským požadavkům rozumí všichni stejně, a aby se případné chyby našly dříve, než je převezme vývojářský tým.

Klíčem k úspěšnému vývoji požadavků je postupné upřesňování. Zkoumání požadavků si naplánujte v cyklech, požadavky na nejvyšší úrovni upřesňujte do podrobností a správnost svých analýz ověřujte s uživateli. Tento proces trvá dlouho a může být celkem únavný, ale při nejisté práci na definici nového softwarového výrobku je zcela nezbytný.

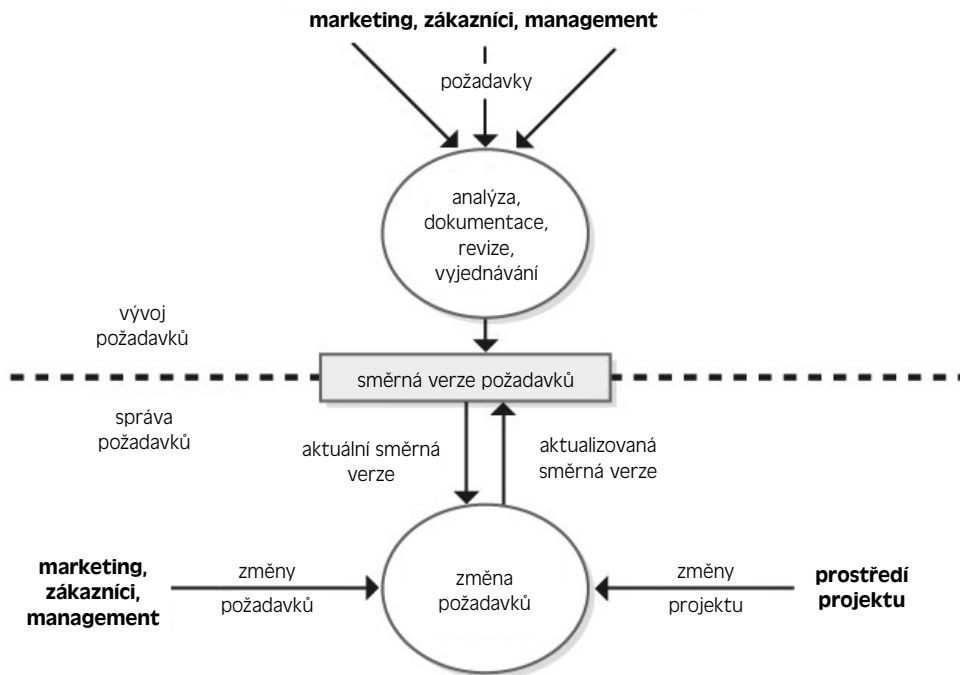
Správa požadavků

Správa požadavků znamená „shodnout se se zákazníkem na požadavcích softwarového projektu a tuto shodu udržovat“ (Paulk a kolektiv 1995). Tato shoda je zakotvena v psané

specifikaci a modelech. Souhlas zákazníka je jen jedna polovina rovnice vedoucí ke schválení požadavků – vývojáři musí specifikace schválit též a musí souhlasit s tím, že podle nich systém postaví. Do správy požadavků patří například následující činnosti:

- Definice směrné podoby požadavků pro danou iteraci projektu.
- Posouzení navrhovaných změn v požadavcích a vyhodnocení pravděpodobných následků každé změny před jejím schválením.
- Řízené zapracování schválených změn do projektu.
- Aktualizace projektových plánů podle požadavků.
- Vyjednávání nových závazků podle očekávaného dopadu změn požadavků.
- Sledování jednotlivých požadavků až k odpovídajícím návrhům, zdrojovému kódu a testovacím scénářům.
- Sledování stavu jednotlivých požadavků a změn v průběhu celého projektu.

Další pohled na rozdíl mezi vývojem a správou požadavků nabízí obrázek 1.3. V této knize najdete několik desítek konkrétních postupů pro sběr, analýzu, specifikaci i správu požadavků.



Obrázek 1.3. Hranice mezi vývojem a správou požadavků.

Každý projekt má své požadavky

Kritickou roli, kterou požadavky hrají v softwarovém projektu, výmluvně popsal Frederick Brooks ve své klasické esejí *No Silver Bullet: Essence and Accidents of Software Engineering* z roku 1987:

Nejtěžší samostatnou fází stavby softwarového systému je rozhodnout, co přesně má vzniknout. Žádná z ostatních částí konceptuální práce není tak složitá, jako vybudování podrobných technických požadavků včetně všech rozhraní k lidem, strojům a dalším softwarovým systémům. Žádná z ostatních částí práce systém tak nezmrzačí, když ji uděláte špatně. Žádná z ostatních částí se tak těžko neopravuje později.

Každý program nebo systém, který software obsahuje, má zlepšit život svým uživatelům. Čas strávený snahou o pochopení potřeb uživatelů je vysoce návratná investice do úspěchu projektu. Pokud vývojáři nemají k dispozici psané požadavky, na kterých se uživatelé shodnou, jak by tyto zákazníci mohli uspokojit? Dokonalé pochopení si vyžadují i požadavky na software, který není určený pro komerční využití – například knihovny, komponenty a nástroje určené pro interní použití vyvojářským týmem.

Často se stává, že se požadavky dají plně specifikovat až po zahájení vývoje. V takových případech zvolte metodu postupného upřesňování – požadavky implementujte po jednotlivých částech a před každou další částí si od zákazníka vyžádejte zpětnou vazbu. Tento přístup ale nesmíte brát jako záminku pro psaní kódu ještě před zamyšlením nad požadavky pro další iteraci. Postupné upřesňování kódu je dražší než postupné upřesňování představ.

Lidé se někdy bojí času, který jim psaní softwarových požadavků zabere, ale psaní požadavků není složité – složité je hledání požadavků. Samotné psaní je převážně záležitost rozepisování, upřesňování a přepisování. Důkladné pochopení požadavků na váš projekt zajistí, že budete pracovat na správném problému a najdete jeho nejlepší řešení. Požadavky vám umožní práci rozdělit podle důležitosti a odhadnout námahu a prostředky, které si projekt vyžádá. Když neznáte všechny požadavky, nemůžete vědět, kdy bude projekt hotový a jestli dosáhne požadovaných cílů. Kdyby přišlo na krácení jeho rozsahu, bez analýzy požadavků nebudete vědět, co si můžete dovolit vypustit.

Z praxe

Žádné předpoklady

Nedávno jsem narazil na vývojářský tým, který pracoval ve vlastním nástroji pro softwarové inženýrství. Součástí tohoto nástroje byl i editor zdrojových kódů – nikoho ovšem, bohužel, ve specifikaci nenapadlo uvést, že by se zdrojové kódy měly dát tisknout, ačkoliv s tím všichni nepochybně počítali. Kdykoliv pak měla proběhnout revize kódu, vývojáři si svůj kód museli přepsat ručně.

Pokud si všechny předpokládané požadavky nedáte na papír, nesmí vás překvapit, když výsledný software nesplní očekávání uživatelů. Pravidelně se sami sebe ptejte, co v danou chvíli předpokládáte, abyste na podobné skryté myšlenky přišli. Když na nějaký předpoklad přijdete během diskuse o požadavcích, запиšte si ho a ověřte si jeho správnost. Pokud svým