

Obsah

Předmluva	11
Poděkování.....	11
O autorech.....	12
Úvodem	12
Komu je tato kniha určena.....	13
Jak byste měli tuto knihu číst.....	13
Web	14

KAPITOLA 1

Úvod do architektury softwaru	15
Použití procesu.....	16
Stručný popis procesu.....	17
Rozsah	21
Shrnutí.....	21

KAPITOLA 2

Architektura, Architekt, Navrhování.....	23
Architektura.....	24
Architektura definuje strukturu.....	25
Architektura definuje chování	26
Architektura se zaměřuje na zásadní prvky.....	27
Architektura vyvažuje potřeby investorů.....	27
Architektura ztělesňuje rozhodnutí, vycházející z podstaty věci.....	28
Architektura může odpovídat určitému stylu navrhování	28
Architektura je ovlivňována svým prostředím	29
Architektura ovlivňuje strukturu vývojového týmu	29
Architektura je přítomna v každém systému.....	30
Architektura má určitý rozsah	30
Architekt	32
Architekt je technickým vedoucím.....	32
Role architekta může být vykonávána týmem.....	33
Architekt rozumí procesu vývoje softwaru.....	34
Architekt má znalosti z obchodní oblasti.....	34

Architekt má technologické znalosti	35
Architekt má návrhové zkušenosti	35
Architekt umí programovat	35
Architekt umí dobře komunikovat	36
Architekt přijímá rozhodnutí	36
Architekt si uvědomuje organizační politiky	36
Architekt je vyjednávačem	37
Navrhování.....	37
Navrhování je vědou	39
Navrhování je uměním	39
Navrhování pokrývá mnoho disciplín	39
Navrhování je nekončící činnost	40
Navrhování je řízeno mnoha investory	41
Navrhování často znamená přijímání kompromisů	41
Navrhování uznává zkušenost	42
Navrhování probíhá jak shora dolů, tak i zdola nahoru	42
Přínosy navrhování	42
Navrhování řeší otázku systémových kvalit	43
Navrhování vede ke shodě	43
Navrhování podporuje proces plánování	43
Navrhování umožňuje zachování integrity návrhu	45
Navrhování pomáhá řídit složitost	45
Navrhování vytváří základ pro opětovnou použitelnost	45
Navrhování snižuje náklady na údržbu	46
Navrhování podporuje analýzu dopadů	46
Shrnutí	47

KAPITOLA 3

Základy metody

49

Klíčové koncepty.....	50
Obsah metody	52
Role	52
Výsledek práce.....	53
Činnost	54
Úkol.....	55
Proces	56
Sériový proces (proces vodopádu)	56
Iterativní proces.....	57
Agilní proces.....	61
Shrnutí	62

KAPITOLA 4

Dokumentace architektury softwaru63

Konec hry	64
Klíčové koncepty	66
Hlediska a pohledy	67
Základní hlediska	68
Příčná hlediska	68
Pohledy a diagramy	70
Přínosy hledisek a pohledů	71
Modely	71
Úrovně realizace	72
Přínosy modelů	73
Charakteristiky rámce popisu architektury	74
Model architektury softwaru, založený na 4+1 pohledech	75
Zachmanův rámec	75
Rozanski a Woods	76
Rámec popisu architektury	78
Hlediska	78
Výsledky práce	81
Úrovně realizace	81
Soulad pohledů	81
Dokument architektury softwaru	83
Shrnutí	83

KAPITOLA 5

Opětovně použitelné prvky architektury85

Zdroje architektury	86
Metamodel prvků architektury	87
Vývojové prvky	88
Běhové prvky	88
Druhy prvků	89
Referenční architektura	89
Vývojová metoda	90
Katalog hledisek	90
Styl navrhování	90
Mechanismus navrhování	91
Vzor	91
Referenční model	94
Existující aplikace	95
Aplikační balík	95
Aplikační rámec	96
Knihovna komponent/komponenta	96
Atributy prvku architektury	96

Další úvahy týkající se opětovné použitelnosti	98
Shrnutí	99

KAPITOLA 6

Úvod do případové studie101

Použití procesu	102
Rozsah případové studie	104
Projektový tým.....	105
Externí vlivy.....	107
Přehled aplikace.....	108
Vize společnosti VašeCesta	110
Popis problému	110
Investoři	110
Funkcionalita.....	112
Kvality	113
Omezení.....	114
Shrnutí	114

KAPITOLA 7

Definice požadavků.....115

Souvislost požadavků s architekturou	118
Funkční a nefunkční požadavky.....	119
Postupy dokumentování požadavků	120
Použití procesu	121
Porozumění popisům úkolů	122
Definice požadavků: přehled činnosti	122
Shrnutí	158

KAPITOLA 8

Tvorba logické architektury159

Přechod od požadavků k řešení	162
Kolik logické architektury?	164
Minimalizace logické architektury	164
Logická architektura jako investice	165
Význam sledovatelnosti	165
Použití procesu	166
Tvorba logické architektury: přehled činnosti	166
Shrnutí	222

KAPITOLA 9

Tvorba fyzické architektury 223

Přechod od logické architektury k fyzické.....	224
Použití procesu.....	227
Tvorba fyzické architektury: přehled činnosti.....	228
Úkol: Průzkum prvků architektury.....	230
Úkol: Definice přehledu architektury	231
Úkol: Dokumentace architektonických rozhodnutí.....	233
Úkol: Přehled funkčních prvků	233
Mapování logických funkčních prvků na fyzické funkční prvky.....	234
Identifikace fyzických funkčních prvků.....	236
Pořízení produktů.....	238
Zohlednění technologicky specifických vzorů.....	239
Úkol: Přehled prvků nasazení.....	245
Mapování logických prvků nasazení na fyzické prvky nasazení	245
Identifikace fyzických prvků nasazení	246
Pořízení hardwaru	248
Úkol: Ověření architektury	248
Úkol: Ukázka realizovatelnosti architektury.....	248
Úkol: Podrobné funkční prvky	249
Úkol: Podrobné prvky nasazení.....	251
Úkol: Validace architektury	254
Úkol: Aktualizace dokumentace architektury softwaru	255
Úkol: Kontrola architektury s investory	255
Shrnutí.....	256

KAPITOLA 10

Za hranicemi základů 257

Architekt a projektový tým	258
Architekt a požadavky	258
Architekt a vývoj	259
Architekt a testování.....	260
Architekt a řízení projektu	261
Architekt a konfigurační management.....	261
Architekt a řízení změn	263
Architekt a vývojové prostředí.....	264
Architekt a obchodní analýza.....	265
Architekt a externí vlivy.....	265
Podniková architektura	266
Vedení návrhu	267
Poskytovatel infrastruktury	268
Poskytovatel údržby a podpory aplikace	268

Navrhování složitých systémů	269
Je nutné vyvinout mnoho rozdílných funkcí	270
Vývoje se účastní mnoho lidí	270
Systém je značně distribuovaný	272
Vývojový tým je distribuovaný	273
Provozní kvality představují extrémní výzvu	274
Jedná se o systém systémů	275
Shrnutí	277
Závěrem: Poznámka od autorů	278

PŘÍLOHA A

Metamodel architektury softwaru279

Definice základních pojmů metamodelu	281
--	-----

PŘÍLOHA B

Katalog hledisek283

Přehled investorů	285
Základní hlediska	286
Příčná hlediska	288
Soulad pohledů	290

PŘÍLOHA C

Přehled metody293

Role	294
Výsledky práce	296
Činnosti	298
Úkoly	299
Činnost: Definice požadavků	299
Činnost: Tvorba logické architektury	302
Činnost: Tvorba fyzické architektury	303
Fáze	303
Vznik	303
Rozpracování	304
Zhotovení	304
Přechod	305

PŘÍLOHA D**Kontrolní seznam požadavků na architekturu..... 307**

Funkční požadavky.....	308
Požadavky použitelnosti.....	309
Požadavky spolehlivosti.....	309
Požadavky výkonu	309
Požadavky podporovatelnosti	310
Omezení	310
Podniková omezení	310
Omezení architektury.....	311
Omezení vývoje.....	311
Fyzická omezení.....	312

Přehled pojmů 313**Odkazy..... 317****Rejstřík 323**

Předmluva

Problematika navrhování softwarových systémů nabývá v současné době čím dál většího významu. Ačkoliv již byla vydána celá řada knih a publikováno mnoho dokumentů, zabývajících se touto problematikou, jen některé z nich se věnují i praktické stránce navrhování softwaru. A tato kniha je jednou z nich.

Jak uvidíte, autoři vám v ní nabízejí svůj zkušený náhled na architekturu: dozvíte se, co architektura softwaru vlastně je, jak se projevuje, kdo ji dělá a jaké jsou její vazby na další části projektů, jejichž cílem je vývoj nějakého softwaru. Ačkoliv tato kniha popisuje všechny důležité základy, představující architekturu systému, její síla spočívá především v hlubokém zkoumání metody navrhování. Říkám *hlubokém*, protože autoři popisují svoje postupy a způsoby pomocí dobře definovaných modelů. Naštěstí to však činí způsobem srozumitelným a dostatečně podrobným, přičemž zvláštní důraz kladou na různé pohledy, vytvářející úplný obrazek architektury systému. Do určité míry se také zabývají překlenutím mezery mezi logickou a fyzickou architekturou systému. Autoři knihy mne také velice potěšili tím, že do ní zahrnuli i otázky lidských zdrojů – především pak role architekta a týmu architektury systému a propojení různých výstupů architektury na činnosti jednotlivých členů týmu.

Na základě poměrně úplné případové studie pak autoři ilustrují využití svých myšlenek v praxi. Jejich popis různých nástrah, jimž by se měl každý softwarový architekt vyhnout, vychází z jejich vlastních zkušeností v oboru vývoje softwaru, a proto je velmi užitečný. Ve skutečnosti toto je jedna z nejužitečnějších knih, neboť v ní najdete myšlenky, které může kterákoliv softwarová firma okamžitě využít v praxi.

Grady Booch

Poděkování

Na vzniku této knihy se spolupodílely dvě klíčové osoby. První z nich byl Grady Booch, který nejenom zaslal sémě, z něhož tato kniha vzešla, ale i laskavě napsal předmluvu, ke své recenzi připojil podrobné komentáře a po celou dobu psaní knihy nás podporoval. Druhou pak je Philippe Kruchten, který nás neustále ohromoval svými anekdotami a zkušenostmi. I Philippe nás podporoval po celou dobu psaní této knihy: přicházel s mnoha nápady, poskytl nám mnoho rozhovorů a napsal recenzi a seznámil nás s mnoha architekty, s nimiž jsme během psaní této knihy hovořili.

Chtěli bychom také poděkovat Bradu Appletonovi, Davu Brainesovi, Alanu Brownovi, Marku Dicksonovi, Celso Gonzalezovi, Holgeru Heussovi, Bobby Higginsovi, Richi Hilliardovi, Kelli Hous-tonovi, Bradu Jacksonovi, Robertu Kitzbergerovi, Colinu Renoufovi, Nicku Rozanskému, Ricku Smithovi a Eoinu Woodsovi za podrobné komentáře, které připojili ke svým recenzím. Díky nim je tato kniha o mnoho lepší.

Mnozí z architektů a projektových manažerů, s nimiž jsme hovořili, věnovali spolupráci s námi značné množství svého času. Jsou to Ian Charters, Philippe Kruchten, Helen McCann, Gordon McClean a Nick Rozanski.

Dále bychom chtěli poděkovat mnohým zákazníkům, spolupracovníkům a známým, kteří ovlivnili naši práci a tedy i tuto knihu. Naneštěstí se jedná o příliš mnoho jednotlivců, takže je tu nemůžeme vyjmenovat všechny; především bychom tedy chtěli poděkovat mnoha našim spolupracovníkům z firmy IBM, účastníkům konferencí WICSA (Working IEEE/IFIP Conference on Software Architecture), členům pracovní skupiny 2.10 (Software Architecture) organizace IFIP (International Federation for Information Processing) a specializované skupině SPA (Software Practice Advancement) společnosti BCS (British Computer Society). Zejména účast na setkáních a konferencích nám umožnila seznámení s význačnými osobnostmi jak z průmyslového, tak i univerzitního prostředí.

A nakonec bychom chtěli poděkovat všem těm z nakladatelství Pearson Education, kteří se zúčastnili tohoto projektu. Především bychom chtěli poděkovat Chrisi Guzikowskému, který pro nakladatelství vyhledává nové knihy, a také Raině Chrobakové či Sheri Cainové a celému produkčnímu týmu.

O autorech

Peter Eeles je vedoucím IT architektem u firmy IBM, pracujícím v rámci divize IBM Software Group ve skupině Rational. Ve své funkci pomáhá firmám zlepšovat jejich procesy při vývoji softwaru, přičemž se zaměřuje a zajímá především o proces navrhování. Peter působí v softwarovém průmyslu od roku 1985 a značnou část své profesionální kariéry strávil navrhováním a následnou implementací rozsáhlých, distribuovaných systémů. Peter je spoluautorem knih *Building J2EE Applications with the Rational Unified Process* (nakladatelství Addison-Wesley, 2002) a *Building Business Objects* (nakladatelství John Wiley & Sons, 1998). Peter je společníkem společnosti BCS (Fellow of British Computer Society) a společníkem organizace IET (Fellow of the Institute of Engineering and Technology), členem technického personálu firmy IBM, certifikovaným IT architektem organizace Open Group a autorizovaným IT specialistou (CITP – Chartered IT Professional).

Peter Cripps je IT architektem, působícím v divizi IBM Global Business Services ve Velké Británii. V softwarovém průmyslu působí od roku 1980. Během své kariéry pracoval jako programátor, softwarový inženýr a procesní inženýr v mnoha odvětvích, včetně telekomunikací, finančních služeb, maloobchodu a vládních organizací. Mezi hlavní oblasti technických zkušeností a zájmu Petera patří využití vývojových postupů, vycházejících z využití komponent a služeb, a návrh postupů pro vývoj dobré architektury. Peter je členem společnosti BCS (Member of British Computer Society) a autorizovaným IT specialistou (CITP – Chartered IT Professional).

Úvodem

Před několika lety si autoři knihy všimli iniciativy *Handbook of Software Architecture*, na jejímž počátku stojí Grady Booch (www.handbookofsoftwarearchitecture.com). Cílem iniciativy Gradyho Boocha je:

Kodifikovat architekturu velkého souhrnu zajímavých softwarových systémů, popsat je způsobem, odhalujícím jejich základní postupy a umožňujícím porovnání mezi různými doménami a styly architektury.

Zatímco Grady se zaměřuje především na výsledné architektury, my jsme se domnívali, že by bylo přinejmenším stejně zajímavé porozumět postupům, používaným úspěšnými softwarovými architekty při tvorbě struktur různých systémů. Naším konečným cílem samozřejmě bylo zopakovat je-

jich úspěch. K dokončení celé této cesty jsme potřebovali několik let. Během té doby bylo realizováno mnoho projektů, hovořili jsme s mnoha architekty a mnoho vývojových metod jsme rozebrali doslova na kousky. Takto získané poznatky přispěly k našemu porozumění podstaty toho, co při navrhování softwarových systémů funguje či naopak nefunguje. Tato kniha je vyvrcholením naší cesty.

Existuje několik vynikajících knih, popisujících určité aspekty procesu navrhování softwaru. Z těchto knih jsme také čerpali. Některé z těchto knih se věnují například dokumentování architektury softwaru, jiné zase jejímu vyhodnocování. Každý z těchto aspektů zapadá do většího celku, neboť každý z nich představuje důležitý prvek procesu navrhování softwaru. Jedním z cílů této knihy tedy je ukázat vám tento větší celek, a to pomocí konsolidovaného pohledu na všechny aspekty navrhování v kontextu typického projektu vývoje softwaru.

Chtěli bychom poznamenat, že tato kniha vám nepředepisuje žádný konkrétní způsob pro vývoj softwaru. Namísto toho popisuje klíčové prvky, s nimiž se pravděpodobně setkáte v kterémkoliv moderním vývojovém prostředí v rámci podpory procesu návrhu.

Komu je tato kniha určena

Je zřejmé, že tato kniha je určena především softwarovým architektům (či těm, kteří se jimi chtějí stát), snažícím se pochopit to, jak jejich role zapadá do celého procesu vývoje softwaru. Může však být užitečná i pro *specializované* architekty, jako například aplikační architekty či architekty zabezpečení. Obecněji vzato tuto knihu může využít každý, kdo chce lépe pochopit roli softwarového architekta. Jako taková pak bude přínosem pro všechny členy vývojového týmu, včetně vývojářů samotných, testerů, obchodních analytiků, projektových manažerů, manažerů konfigurace a procesních inženýrů. V neposlední řadě bude tato kniha zajímavá i pro vysokoškolské studenty, kteří chtějí získat představu o stále důležitější roli softwarových architektů během procesu vývoje nějakého softwaru.

Jak byste měli tuto knihu číst

Celá kniha je zhruba rozdělena do tří částí.

První část, tvořená kapitolami 1 až 5, obsahuje souhrnný přehled základních konceptů architektury, architekta a navrhování, dokumentování architektury softwaru a opakovaně použitelných prvků architektury.

Druhá část sestává z kapitol 6 až 9 a její hlavní náplní jsou kapitoly, *související s případovou studií*. Lze říci, že tyto kapitoly vás na základě ukázkové aplikace provedou typickým projektem vývoje softwaru, přičemž vaše pozornost bude směřována především k roli architekta. Tyto kapitoly byly napsány tak, aby vám umožnily rychlé čtení a snadné vyhledání těch témat, která vás zajímají. Každá kapitola, související s případovou studií, je uspořádána především podle úkolů a činností. Při psaní těchto kapitol jsme použili několik stylů a konvencí. Především platí, že veškeré odkazy na prvky procesu, jako například úkoly, výsledky práce a role, jsou zvýrazněny tučným písmem. Příkladem může být odkaz **Dokumentace architektury softwaru**.

V poslední části pak najdete kapitolu 10, obsahující různá další témata a zabývající se možnostmi využití konceptů, popsanych v předcházejících kapitolách, při navrhování složitých systémů.

V celé knize se také setkáte s užitečnými poznámkami, odlišenými šedým pozadím, které lze rozdělit do těchto kategorií:

- ◆ **Koncepty** vás seznamují s myšlenkami či celými sadami myšlenek, vztahujících se k právě popisovaným tématům.
- ◆ **Kontrolní seznamy** obsahují užitečné seznamy bodů, které lze během provádění určitého úkolu kontrolovat.
- ◆ **Doporučení** vás seznamují s efektivními přístupy, ověřenými a vyzkoušenými v praxi.
- ◆ **Úskalí** obsahují informace o postupech, jimž byste se měli v ideálním případě zcela vyhnout, neboť jejich výsledky jsou vždy negativní.

K popisu určitých aspektů architektury jsme při psaní této knihy do značné míry využili Unified Modeling Language (UML). Všechny UML diagramy byly vytvořeny pomocí IBM Rational Software Architect.

Web

Vydání této knihy bylo provázeno zpřístupněním webu *processofsoftwarearchitecting.com*, na němž čtenáři najdou dodatečné informace a jehož prostřednictvím mohou i komunikovat s autory.