

Vzťah zákazník a vývojár v softvérových projektoch

IVAN ŠKOVRAŇ

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
skovran@ynet.sk*

Abstrakt. Počas celého trvania vývoja softvérového systému dochádza ku komunikácii medzi zákazníkom a vývojárom. Podmienkou úspešnej špecifikácie požiadaviek, návrhu systému, predvážania prototypu, či testovania je dobrý a fungujúci vzťah medzi zákazníkom a vývojárom. Tvorba softvérového systému nie je jednoduchá úloha, preto by sa obidve strany mali snažiť spraviť čo najviac pre úspešnú spoluprácu. Úlohou zákazníka je čo najlepšie, najpresnejšie a najrozumiteľnejšie vysvetliť vývojárovi svoje požiadavky na softvérový systém. Na druhej strane, vývojár musí vedieť počúvať a pochopiť, čo sa mu zákazník snaží vysvetliť. Na základe získaných informácií musí vývojár navrhnuť softvérový systém, ktorý bude vyhovovať požiadavkám používateľa. V tejto práci sa snažíme zachytiť základné aspekty vo vzťahu zákazník a vývojár pri tvorbe softvérového systému. Najprv oboznámime čitateľa s problémovým prostredím. Potom uvedieme hlavné prekážky a problémy vo vzťahu zákazníka a vývojára. V poslednej kapitole sa venujeme rôznym metódam a nástrojom používaným pri tvorbe softvérového systému, so zameraním na zákazníka a vývojára.

Úvod

Proces získavania požiadaviek pri návrhu softvérového systému je nerozlučne spätý so získavaním najrozmanitejších informácií od zákazníka. V tomto procese sa vývojár snaží za pomoci rôznych nástrojov a postupov čo najlepšie porozumieť požiadavkám zákazníka. To, aký úspešný bude návrh softvérového systému, a v konečnom dôsledku aj aký úspešný bude celý softvérový projekt, závisí na komunikácii a spolupráci zákazníka a vývojára. Problém získavania požiadaviek so sebou nesie okrem aspektov spätých s technickou odbornosťou aj aspekty personálne, kultúrne, či dokonca psychologické. V každom prípade sa nejedná o triviálny problém. Aby bol vývojár schopný porozumieť požiadavkám a potrebám zákazníka, potrebuje vedieť, ako so zákazníkom pracovať a komunikovať tak, aby ho úspešne zapojil do procesu tvorby softvéru.

Keď hovoríme o komunikácii medzi používateľom a vývojárom, máme na mysli tak komunikáciu medzi tímom vývojárov a komunitou používateľov, ako aj osobnú komunikáciu medzi jedným vývojárom a jedným používateľom. Vďaka osobnej a bezprostrednej komunikácii s jednotlivými zákazníkmi a sledovaniu a porozumeniu práce zákazníka je vývojár, a prostredníctvom neho celý vývojový tím, schopný dokonale porozumieť požiadavkám kladeným na softvérový systém. Samozrejme, komunikácia medzi nimi môže prebiehať aj nepriamo, prostredníctvom rôznych nástrojov na to určených, alebo prostredníctvom tretej osoby. Je to však práve osobný rozhovor tvárou v tvár, ktorým sa dá najlepšie odhaliť skutočná podoba požiadaviek zákazníka. Dobře vedený osobný rozhovor dokáže často odhaliť vývojárovi a dokonca aj zákazníkovi to, čo sa vlastne od systému očakáva, a ktoré črty systému sú pre zákazníka najpodstatnejšie.

V tejto práci sa bližšie pozrieme na súčasné možnosti a trendy v procese získavania požiadaviek. Zhrnieme základné metódy a postupy a pozrieme sa na hlavné funkcie a úlohy osôb, ktoré sú potrebné pre úspešné získavanie požiadaviek. Podrobnejšie sa budeme venovať problematike úzkej spolupráce medzi zákazníkom a vývojárom a možným spôsobom ich komunikácie. Možno sa pokúsime aj osloviť vývojárov a zákazníkov a poradiť im, ako zlepšiť ich vzájomnú spoluprácu, ktorej cieľom bude vytvorenie softvérového produktu, s ktorým budú obe strany spokojné.

Problémové prostredie

Prvým krokom pri vývoji softvérového systému je zistiť, na čo má byť systém určený, čo presne má robiť. Proces získavania požiadaviek má niekoľko fáz. Najskôr je potrebné identifikovať požiadavky od používateľa. Ďalším krokom je analýza týchto požiadaviek a ich následné zdokumentovanie za účelom špecifikácie a verifikácie používateľom. Získavanie požiadaviek môžeme klasifikovať ako úspešné, ak má zákazník pocit, že vývojár porozumel všetkým jeho potrebám a požiadavkám. Z pohľadu vývojára je získavanie požiadaviek úspešné, ak je pomocou nich možné dobre navrhnuť informačný systém.

Na vytváraní špecifikácie pre informačný systém väčšinou pracuje a spolupracuje väčší počet ľudí. Preto je potrebné pre každého člena zapojeného do projektu konkrétne špecifikovať jeho funkcie a role. Podľa [4] môžeme role a funkcie klasifikovať nasledovne:

- *Kupec* je človek, ktorý má na starosti zmluvy, účty a peniaze. Kupec sa priamo nezúčastňuje na tvorbe projektu, jeho úloha je zviazaná skôr s manažmentom procesu.
- *Koncový používateľ* bude vytváraný softvérový systém používať. Títo ľudia by sa mali aktívne zúčastňovať na tvorbe softvérového systému, hlavne vo fáze špecifikácie požiadaviek.

- *Odborník z doménovej oblasti* najlepšie rozumie prostrediu v ktorom bude softvérový systém nasadený, hlavne z pohľadu systémových požiadaviek.
- *Údržbár* je zodpovedný za riadenie zmien a implementáciu zmien v softvérových produktoch, ktoré sa už v praxi používajú.
- *Programový manažér* je človek zodpovedný za obchodné a trhové záležitosti a vykonáva dozor nad vývojom softvéru. Často ide o ľudí, ktorí priamo komunikujú so zákazníkom.
- *Inžinier požiadaviek* je vlastne systémový inžinier a jeho úlohou je identifikácia požiadaviek a ich zdokumentovanie.
- *Softvérový inžinier* je expertom na návrh systému, vytvorenie prototypu a ďalšie technické záležitosti.
- *Tester* je zodpovedný za vývoj a vykonávanie testov, ktoré majú overiť správnosť a preukázať schopnosti systému. Ich činnosť je dosť dôležitá, pretože zákazníkovi musíme nejako preukázať, že systém spĺňa jeho požiadavky.

Prekážky vo vzťahu zákazník a vývojár

Proces získavania požiadaviek a s tým spojená komunikácia zákazníka a vývojára so sebou prináša rôzne problémy. Tu sa pokúsime zhrnúť základné problémy, ktoré sa pri tomto zložitom procese môžu vyskytnúť.

Slabá komunikácia

Vďaka komunikácii o cieľoch, úlohách, plánoch, obmedzeniach a prioritách nášho systému môžeme vyvinúť softvérový systém, ktorý bude vyhovovať potrebám používateľa, no nemusí splniť všetky jeho očakávania. Pri komunikácii totižto musíme pozerieť nie len na to o čom komunikujeme, ale aj na to, ako komunikujeme.

Najlepším a najefektívnejším spôsobom komunikácie je rozhovor tvárou v tvár. Pri tomto spôsobe komunikácie musia zúčastnení vedieť vhodne verbálne vyjadriť svoje požiadavky a musia byť samozrejme odborníkmi v téme rozhovoru. Inak je takmer isté, že nastanú problémy. Ak pri rozhovore medzi vývojárom a zákazníkom jedna strana zistí, že druhá nepočúva a nejaví o rozhovor záujem, znižuje sa tak efektivita tejto spolupráce a komunikácia viazne. Naopak, ak je vývojár dobrým poslucháčom, povzburdený rozprávač si to všimne, čo môže viesť len ku zlepšeniu komunikácie. Niektoré z príznakov, že s komunikáciou nie je niečo v poriadku:

- osoba na to, čo hovoríte, vôbec nereaguje
- ak s vami ten druhý stále súhlasí (za predpokladu, že nie ste jeho šéf :)
- ak druhá osoba nie je nikdy spokojná s tým, čo poviete.

Medzi vývojárom a zákazníkom funguje aj nepriama komunikácia, tj. komunikácia pomocou rôznych nástrojov alebo prostredníctvom tretej osoby. Je takmer vylúčené, aby medzi oboma stranami existoval len tento spôsob komunikácie. Tretia osoba často nepozná presné potreby a požiadavky zákazníka, a tak často dochádza k ich prekrúteniu a skresleniu. Nepriama komunikácia je vhodná len ako doplnujúca, a to v prípade, že osobný kontakt nie je možný.

Nezrozumiteľnosť a neodbornosť

Častou prekážkou pri komunikácii na technickej úrovni je používanie odborných názvov, termínov, ktorým niektorá zo strán nerozumie. Preto by sa mala strana, ktorá používa tieto termíny, snažiť svoje vyjadrovanie prispôsobiť úrovni druhej strany a snažiť sa zrozumiteľným jazykom vysvetliť daný problém.

Ďalším problémom, ktorý s touto tematikou súvisí je to, keď sa vývojár správa ku zákazníkovi ako ku hlupákovi, ktorý sa „do toho“ vôbec nerozumie. Takýmto prístupom nič nedosiahneme, toto často urážlivé správanie nikam nevedie. Vývojár by mal naopak zákazníkovi pomôcť porozumieť problematike ktorej nerozumie. Mal by pochopiť, že aj človek, ktorý sa vďaka stlačenému Caps Lock-u, nevie prihlásiť do systému, môže byť vo svojej oblasti skutočným odborníkom, a že nemá najmenší dôvod sa mu za to posmievať.

Rozdielne oblasti pôsobenia

Väčšina vývojárov má pri vývoji softvérového systému len veľmi málo, resp. žiadne skúsenosti z oblasti, pre ktorú je softvérový systém určený. Vývojárom chýbajú praktické skúsenosti a znalosti, ktoré sa dajú získať len prácou v danom odbore. Tieto skúsenosti má tvorcovi softvérového systému priblížiť samotný zákazník, používateľ. Zo skúsenosti vieme, že vo veľkom množstve vytvorených softvérových systémov, kde sa táto spolupráca podcenila, dostal zákazník hotový produkt, ktorý síce bol vytvorený podľa jeho požiadaviek, ale nenaplnil všetky jeho očakávania.

Na druhej strane je pre zákazníka a používateľa tvorba softvérového systému veľkou neznámou. Zákazník teda nevie, ako musí svoje požiadavky špecifikovať a aké obmedzenia si potrebuje stanoviť, aby sa jeho systém vôbec dal reálne navrhnuť a implementovať. Preto je práve na vývojároch, aby poučili zákazníka o tom, čo je, a čo nie je možné za daných podmienok navrhnuť a vytvoriť. Vývojár musí vedieť ako usmerniť zákazníka správnym smerom, ale tak, aby pôvodné požiadavky zákazníka zostali v najväčšej možnej miere zachované.

Metódy a nástroje

V tejto časti sa budeme venovať rôznym metódam a použitiu rôznych nástrojov, ktorých cieľom je zlepšiť vzťah a komunikáciu medzi zákazníkom a vývojárom. Zároveň uvedieme niektoré príklady z praxe, kde sa takéto techniky a nástroje používajú.

Získavanie informácií za pomoci komunikačných zručností

Ak ako vývojár dokážeme dať zákazníkovi najavo, že ho pozorne počúvame, pomôže to oboj stranám pri získavaní informácií a celkovo pri budovaní ich vzťahu. Prejav ako očný kontakt, poloha tela a výraz tváre pri počúvaní prezradí veľa o tom, či v skutočnosti dávame pozor a počúvame. To ako reagujeme na rozprávajúceho je aspoň tak dôležité ako to, ako rozprávame. Samozrejme, nemali by sme nechať zákazníka rozprávať monologicky. Pýtanie sa otázok, utvrdenie sa v problematike a písanie si poznámok dá jasne najavo, že sme v komunikácii aktívni. Samozrejme, všetkého veľa škodí. S kladením otázok by sme to nemali preháňať a nemali by sme rozprávajúceho stále prerušovať.

Ak zákazník nevie presne čo potrebuje, resp. nevie to definovať, je na vývojárovi, aby mu v tom pomohol. Zákazník používateľ ani nemusí byť schopný presne popísať to, čo potrebuje. Stačí, ak vývojárovi ukáže svoju každodennú činnosť, na ktorú sa viaže vyvíjaný informačný systém. Je úlohou vývojára pomocou týchto informácií navrhnúť pre používateľa konkrétne riešenie.

Naomi Kartenová v knihe *Managing Expectation* [2] ponúka zopár rád, ako získavať informácie od používateľa:

1. Nepredpokladajte čo zákazník myslí. Opakujte otázky, ktoré ste sa už pýtali, preformulujte ich. Dostanete na nich rôzne odpovede z rôznych pohľadov.
2. Žiadajte si vysvetlenie. Buďte si istý, že rozumiете tomu, čo vám zákazník hovorí. Nebojte sa priznať, že niečo nevíete. Zdôraznite zákazníkovi, že vaše otázky sú nevyhnutné k tomu, aby ste plne pochopili jeho očakávania.
3. Zbierajte informácie z viacerých zdrojov. Pýtaním sa rovnakých otázok rôznych ľudí získate širší pohľad na ich potreby. Odpovede od viacerých ľudí vám pomôžu vyplniť diery v informácii získanej od jedného človeka.
4. Dávajte si pozor na nepresnosti. Zvážte zručnosti a reakcie zákazníkov. Odpovedajú vám na otázky z ich vlastného pohľadu, alebo sa vám len snažia povedať to, čo chcete počuť?

Návrhár ako zákazníkovo učenie

V práci *Apprenticing With the Customer* [1] autori hľadajú odpoveď na otázku, aký vzťah by sa mal vytvoriť medzi zákazníkom a vývojárom, aby vývojár čo najlepšie porozumel požiadavkám zákazníka. Ich odpoveď na túto otázku znie: „Tak, ako sa učenie učí zručnosti od svojho majstra, tak sa aj návrhár chce naučiť niečo od zákazníka sledovaním jeho práce. Základné aspekty tohto vzťahu môžeme zhrnúť nasledovne:

- Z pohľadu zákazníka, majstra, nie je schopnosť učiť potrebná. Majster neučí svojho učňa ako klasický učiteľ, ale učí ho vykonávaním svojej práce a komentovaním toho, čo robí. Zákazník teda len vykonáva svoju každodennú činnosť a komunikuje s učňom. Učenie počúva.
- Pozorovanie práce majstra odhalí to, čo je podstatné. Niektoré činnosti sú výsledkom skúseností, iné sa stali zvykom. Najlepšie sa o tom čo, ako, a prečo robím hovorí priamo, ak to v tom okamihu skutočne robím.

- Pozorovanie práce majstra odhalí všetky detaily. Komentovanie priamo pri vykonávaní práce zabráni zovšeobecňovaniu zo strany majstra. Ak majster hovorí o svojej činnosti, a popri tom ju nevykonáva, často sa stane, že si nepamätá všetky detaily, mieša činnosti dokopy, a neopisuje činnosti v správnom poradí.
- Odhalenie štruktúry práce zákazníka. Vývojár sa učí z pozorovania majstra, sleduje jeho metódy a stratégie a snaží sa im porozumieť. Keď im porozumie, začne vytvárať možné návrhy pre systém.
- Učeň sa učí zo skúseností svojho majstra. Majster vie pomocou vlastnej skúsenosti presne povedať, čo ide robiť, čo pri tejto práci potrebuje, aká je postupnosť krokov ktorú vykonáva, čo pomáha učňovi lepšie porozumieť jeho práci.

Medzi učňami a vývojármi však existuje jeden podstatný rozdiel. Učeň sa len naučí vykonávať majstrovu činnosť, vývojár musí navyše navrhnúť systém, ktorý majstrovi zjednoduší jeho činnosť.

- Návrhár musí odhaliť dôležité štruktúru v práci majstra. Návrhár hľadá v jeho práci postupy, obmedzenia a vzory a pod.
- Návrhár si musí ujasniť, či problematike skutočne rozumie. Pokiaľ je návrhár len pasívny, len pozoruje a nekomunikuje, nemôže navrhnúť dobrý systém. Návrhár si musí u používateľa overiť, či problému dobre rozumie.
- Prácou návrhára je zlepšiť prácu majstra. Návrhár musí stále rozmýšľať nad spôsobmi, ako vylepšiť a zefektívniť jeho činnosť.
- Návrhár vie majstra usmerniť. Návrhár je odborníkom na zovšeobecňovanie a vie odhaliť štruktúru práce, a preto musí vedieť majstrovi poradiť a vhodne ho usmerniť.

Prototypovanie

Dobrá spolupráca medzi vývojárom a zákazníkom je dôležitou podmienkou pre úspešné zvládnutie prototypovania. Prototypovanie znamená vytvorenie jednoduchého modelu systému za účelom lepšieho porozumenia systému a požiadavkám. Prototypovanie znižuje riziká a pomáha už na začiatku vývoja systému zabrániť chybám, ktorých odhalenie a riešenie v neskorších fázach vývoja by bolo oveľa náročnejšie a nákladnejšie. Prototypovanie by malo prebiehať nasledovne:

1. Navrhni používateľské rozhranie, len so základnými funkciami.
2. Čo najrýchlejšie vytvor „hrubý“ systém. Systém nemusí byť spoľahlivý, má však mať čo najlepšie vytvorené používateľské rozhranie.
3. Nechaj používateľa „pohrať sa“ s prototypom, čím od neho získaš hneď na začiatku odozvu, kladnú, či zápornú.
4. Pozoruj používateľa. Všimni si, ako používateľ na systém reaguje.

5. Zisti názor používateľa na prototyp. Pomocou vhodných otázok zisti od viacerých používateľov ich názor, požiadavky na zmeny v prototypu a pod.
6. Na základe spätnej väzby od používateľov prerob vytvorený prototyp a vráť sa k prvému kroku.

Pri prototypovaní dochádza k častému a opakovanému kontaktu medzi vývojárom a zákazníkom. Pre úspešný priebeh prototypovania je preto výhodné využiť niektoré zo skôr uvedených informácií o osobnej komunikácii.

Spôsoby výmeny informácií

Autori Mark Keil a Erran Carmel sa problematike vzťahu zákazníka a vývojára venujú z trocha iného pohľadu. Vo svojej práci Customer-Developer Links [3] skúmajú, ako závisí úspešnosť softvérových projektov od počtu a rôznorodosti komunikačných kanálov a techník (liniek) medzi vývojármi a zákazníkmi.

Autori porovnávajú výhody a nevýhody tých ktorých liniek používaných vo vývoji generického softvéru a softvéru na zákazku. Linky rozdeľujú na priame a nepriame. Priame linky sú tie, pri ktorých dochádza ku priamemu kontakty zákazníka a vývojára. Komunikácia cez tretiu osobu, nepriamo, je zasa charakteristická pre nepriame linky. Z prieskumu, ktorý bol vykonaný vyplynuli nasledovné závery:

Čím viac liniek, tým lepšie, ale ...

Projekty, v ktorých bolo použité väčšie množstvo liniek boli spravidla úspešnejšie. Pri úspešných projektoch stredná hodnota počtu použitých liniek bola 5,4. Menej úspešné projekty mali strednú hodnotu počtu liniek na úrovni 3,2.

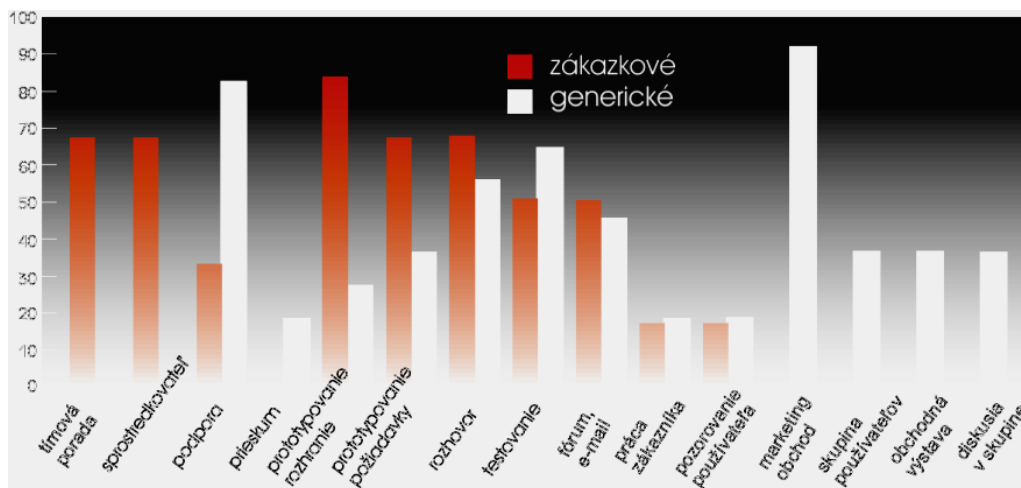
Pravidlo čím viac liniek, tým lepšie, neplatí úplne. Mať v projekte príliš veľa liniek taktiež nie je optimálne. Výdavky spojené s implementáciou veľkého počtu liniek by mohli prevýšiť výhody, ktoré väčšie množstvo liniek prináša, čo by bolo kontraproduktívne.

Nespoliehajte sa na nepriame linky

Na základe výsledkov prieskumu je jasné, že projekty, v ktorých boli vo väčšej miere zastúpené priame linky dopadli úspešnejšie, ako tie s malým počtom priamych liniek. Desať zo štrnástich menej úspešných projektov malo medzi zákazníkmi a vývojármi len jednu, alebo žiadnu priamu linku. Dôvody sú zrejmé. Komunikácia cez tretiu osobu je „zašumená“, sprostredkovateľ nemusí informácie podať tak, ako mu boli zdedené.

Zvážte použitie rôznorodých liniek

Pri vývoji softvéru by manažéri mali zvážiť aj použitie liniek, ktoré nie sú v ich prostredí typicky používané, ale sú vnímané ako vcelku efektívne. Na nasledujúcom obrázku môžeme vidieť, ktoré linky a v akej miere sa používajú pri vývoji generického softvéru a softvéru na zákazku.



Obr.1. Percentuálne zastúpenie liniek pre zákazkové a generické softvérové systémy

Záver

Vzťah zákazník a vývojár v softvérových projektoch. Tento názov reprezentuje problematiku, ktorá v sebe zahŕňa množstvo zaujímavých a rozmanitých úloh. Snaha o ich úspešné riešenie stojí pred každým, kto to myslí s tvorbou softvérových systémov vážne. To, že táto problematika nie je triviálna, si po prečítaní tejto eseje určite každý uvedomil. Zachytáva skutočne široké spektrum tém a úloh, ktorých zvládnutie si vyžaduje veľa odborných znalostí od softvérového inžinierstva, cez komunikačné schopnosti, až po psychológiu.

V eseji som sa snažil načrtnúť, čo všetko táto bohatá téma obsahuje. Podrobne sa zaoberať všetkými jej aspektmi by bolo v takejto krátkej práci nemožné. Preto som sa snažil len zhrnúť o čo vlastne ide, vytvoriť o problematike ucelený obraz. Niektoré časti som rozvinul podrobnejšie, iné som len okrajovo spomenul. Dúfam, že sa aj čitateľovi, ktorému softvérové inžinierstvo veľa nehovorí, podarilo porozumieť aspoň čo-to zo zložitého procesu tvorby informačného systému.

Použitá literatúra

1. Beyer, R. H, Holzblatt, K.: Apprenticing With the Customer. *Communications of the ACM*, Vol. 38, No. 5 (May 1995) 45-52.
2. Karten, N., *Managing Expectations*, Dorset House Publishing, New York, NY, 1994.

3. Keil, M., Carmel, E.: Customer-Developer Links in Software Development. *Communications of the ACM*, Vol. 38, No. 5 (May 1995) 33-44.
4. Saiedian, H., Dale, R.: Requirements engineering: making the connection between the software developer and customer. In: *Information and Software Technology*, Elsevier Science B. V. (2000), 420-428.

Annotation

Customer-Developer Relationship in Software Projects

For the whole software development process it is typical that there is active communication between customer and software developer. Functioning relationship between customer and developer is the key aspect of successful development process including requirements engineering, designing of the system, prototyping and testing. It is not an easy task to design and develop software system. Therefore they have to try hard to create a successful relationship full of collaboration. Basic task of customer is to explain to a developer, as good as it gets, how to fulfill his needs and requirements. On the other side, developer's task is to listen carefully and to understand customer's needs. After that, he must design software system, with the use of gathered information. In this essay we try to explain basic aspects in customer-developer relationship. In the first chapter, we explain to the reader background of this relationship. The next one deals with difficulties. The last chapter is dedicated to techniques and tools for enhancing the elicitation process.