

Implementační balíček

Analýza požadavků na software

Základní profil

Poznámky

Tento dokument je duševním vlastnictvím autorovy společnosti. Informace obsažené v tomto dokumentu však mohou být volně používány. Šíření tohoto dokumentu nebo jeho částí je pro nekomerční využití povoleno, pokud je uvedeno následující oznámení o právech

© Centre d'Excellence en Technologies de l'Information et de la Communication and École de Technologie Supérieure

Komerční využití tohoto dokumentu je přísně zakázáno. Tento dokument je rozšiřován za účelem zlepšení výměny technických a vědeckých informací.

Tento dokument je poskytován tak, jak je. Autor neposkytuje (autoři neposkytují) záruky jakéhokoli druhu, ať už vyjádřené nebo předpokládané, a to v jakékoli záležitosti, včetně, ale nejen pouze, záruky vhodnosti pro obchodovatelnost, záruky exkluzivity nebo záruk za výsledky získané s použitím tohoto dokumentu.

Procesy popsané v tomto implementačním balíčku nemají zamezit použití nebo odradit od použití jakýchkoli dodatečných procesů, které velmi malé podniky mohou považovat za užitečné.

Autoři	S. ALEXANDRE, Centre d'Excellence en Technologies de l'Information et de la Communication (CETIC), Belgie C. Y. LAPORTE, École de Technologie Supérieure (ETS), Kanada
Editoři	C. Y. LAPORTE, École de Technologie Supérieure ANA VAZQUEZ, 5th level, Mexiko
Vytvořeno dne	07.08.2007
Poslední aktualizace	07.07.2009
Stav	Návrh
Verze	1.2
Překlad	Tento překlad byl připraven v rámci zpracování semestrální práce v předmětu 4IT421 <i>Zlepšování procesů budování IS</i> na Vysoké škole ekonomické v Praze.

Verze 1.2

Verze

Datum	Verze	Autor	Změna
07.08.2007	0.1	S. ALEXANDRE	Vytvoření dokumentu
20.08.2007	0.2	C.Y. LAPORTE	Připomínky ke struktuře dokumentu
01.10.2007	0.3	S. ALEXANDRE	Zpracování poznámek a dokončení V1.0
01.10.2007	0.4	S. ALEXANDRE	Připomínky po přezkoumání
08.10.2007	0.5	S. ALEXANDRE	Aktualizace a doplnění části 3.1
14.10.2007	0.6	S. ALEXANDRE – C.Y. LAPORTE	Aktualizace a revize části 3.1
19.10.2007	0.7	S. ALEXANDRE	Aktualizace částí 3.1 a 3.2
29.10.2007	0.8	S. ALEXANDRE	Nahrazení pojmu 'practice' pojmem 'activity' pro soulad s názvoslovím OGF SPEM
02.11.2007	0.9	S. ALEXANDRE	Uspořádání obsahu podle ISO/IEC 12207:2008
27.11.2007	0.10	S. ALEXANDRE	Aktualizace grafického znázornění kroků
01.12.2007	1.0	S. ALEXANDRE	Konečná verze – připravena pro závěrečné posouzení
21.01.2008	1.1	S. ALEXANDRE	Aktualizace matice pokrytí
07.07.2009	1.2	C.Y. LAPORTE	Úprava ustanovení o autorských právech, definic a akronymů; uplatnění nové předlohy DP

Zkratky/Akronymy

Zkr./Akr.	Definice
DP	Implementační balíček (<i>Deployment Package</i> , DP) – sada artefaktů vytvořených pro podporu realizace postupů vybraného rámce ve velmi malém podniku.
VSE	Velmi malý podnik (<i>Very Small Entity</i> , VSE) – v rámci tohoto implementačního balíčku tento pojem označuje podniky, organizace, oddělení (útvary) nebo projekty do 25 osob.

Obsah	
1. Technický popis.....	4
<i>Účel dokumentu</i>	<i>4</i>
<i>Proč je řízení požadavků důležité?.....</i>	<i>4</i>
<i>Hlavní příčiny úspěšnosti IT projektů</i>	<i>5</i>
2. Pojmy a definice.....	7
<i>Obecné pojmy.....</i>	<i>7</i>
<i>Specifické pojmy</i>	<i>7</i>
3. Vztah k ISO/IEC 29110	9
4. Popis procesů, činností, úkolů, kroků, rolí a produktů	11
<i>Úkoly.....</i>	<i>11</i>
<i>Určení požadavků</i>	<i>11</i>
<i>Upřesnění a analýza požadavků</i>	<i>13</i>
<i>Verifikace (ověření) a validace požadavků</i>	<i>14</i>
<i>Řízení změn požadavků</i>	<i>15</i>
<i>Role</i>	<i>16</i>
<i>Artefakty.....</i>	<i>16</i>
5. Šablona	17
6. Příklad postupu činností.....	22
<i>Postup činností analýzy požadavků – příklad 1</i>	<i>22</i>
<i>Postup činností analýzy požadavků – příklad 2</i>	<i>23</i>
7. Kontrolní seznam	24
<i>Kontrolní seznam požadavků.....</i>	<i>24</i>
8. Nástroje	25
<i>Nástroj pro sledovatelnost</i>	<i>25</i>
9. Odkazy na jiné standardy a modely.....	27
<i>Matice odkazů k ISO 9001</i>	<i>27</i>
<i>Matice odkazů k ISO/IEC 12207</i>	<i>28</i>
<i>Matice odkazů pro CMMI.....</i>	<i>28</i>
10. Odkazy	29
11. Hodnoticí formulář	30

1. Technický popis

Účel dokumentu

Tento implementační balíček (Deployment Package, DP) podporuje Základní profil (Basic Profile) definovaný v ISO/IEC 29110, část 5-1: Management and Engineering Guide. DP je sada artefaktů, které byly vytvořeny, aby usnadnily realizaci souboru praktik ve velmi malém podniku (VSE). DP není referenčním modelem procesů – není normativní. Prvky typického DP jsou: popis procesů, činností, úkolů, rolí a produktů, šablona či předloha, kontrolní seznam, příklad, odkazy na standardy či modely a nástroje.

V projektech IT je velmi důležité co nejjednoznačněji definovat požadavky zákazníka, aby se zajistilo, že jim zainteresované strany společně porozumí, a zaručit, že postupný vývoj požadavků je řešen jako součást projektu.

Proces analýzy požadavků zahrnuje vytvoření a spravování *Specifikace požadavků na software* na základě potřeb zákazníka a změn těchto potřeb. Specifikace požadavků na software poté tvoří základ pro rozpočet nákladů, plánování, implementaci a sledování činností v průběhu projektu.

Řízení požadavků je jedním ze základních předpokladů stabilizace procesu a jeho úspěšné opakovatelnosti.

Obsah tohoto dokumentu je zcela *informativní*.

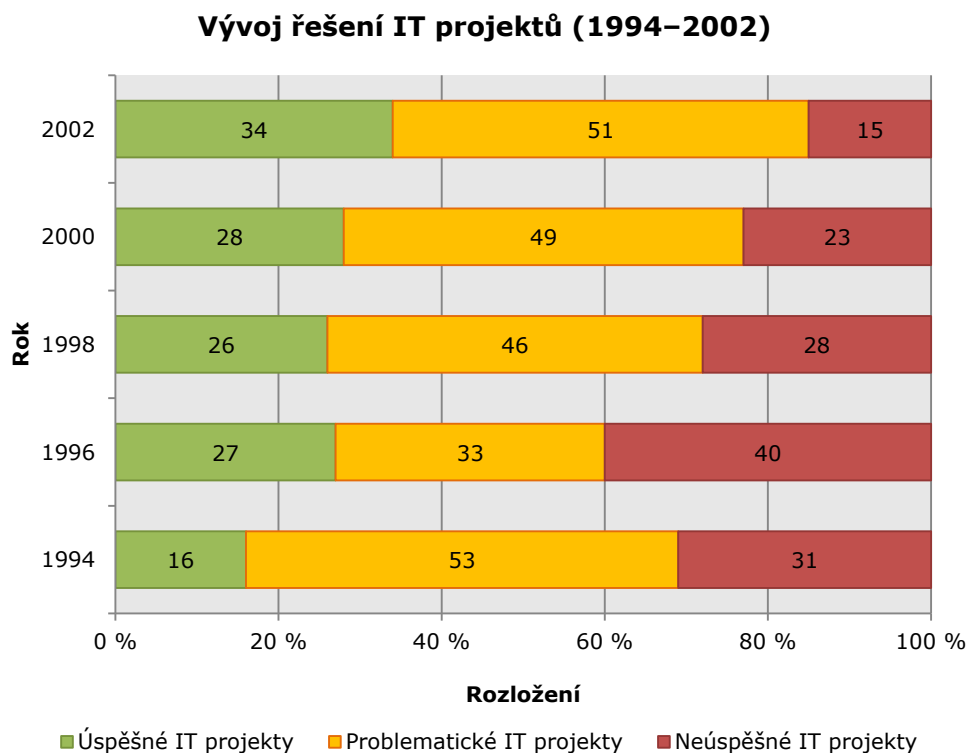
Tento dokument připravili CETIC (Centre of Excellence in Information and Communication Technologies – www.cetic.be), CRPHT (Public Research Centre Henri Tudor's – www.tudor.lu) a ETS (Ecole de Technologie Supérieure – www.etsmtl.ca) v rámci jejich oficiální účasti v pracovní skupině ISO JTC1/SC7/WG24.

Proč je řízení požadavků důležité?

Význam řízení požadavků v softwarovém inženýrství zřetelně zdůraznilo již několik studií. Studie Chaos Report, vydávaná Standish Group (www.standishgroup.com) od roku 1994, analyzovala tisíce IT projektů z celého světa.

Standish Group IT projekty zařadila do tří kategorií řešení

- *Úspěšné* – projekt se všemi původně specifikovanými vlastnostmi a funkcemi je dokončen včas a při splnění rozpočtu.
- *Problematické* – projekt je dokončený a provozuschopný, ale překročil rozpočet, překročil časové odhady, a přitom řeší menší počet vlastností nebo funkcí, než bylo specifikováno při zahájení projektu.
- *Neúspěšné* – projekt je zrušen ještě před dokončením, nebo jeho řešení nikdy nebylo implementováno.



Obrázek 1 Vývoj řešení IT projektů

Hlavní příčiny úspěšnosti IT projektů

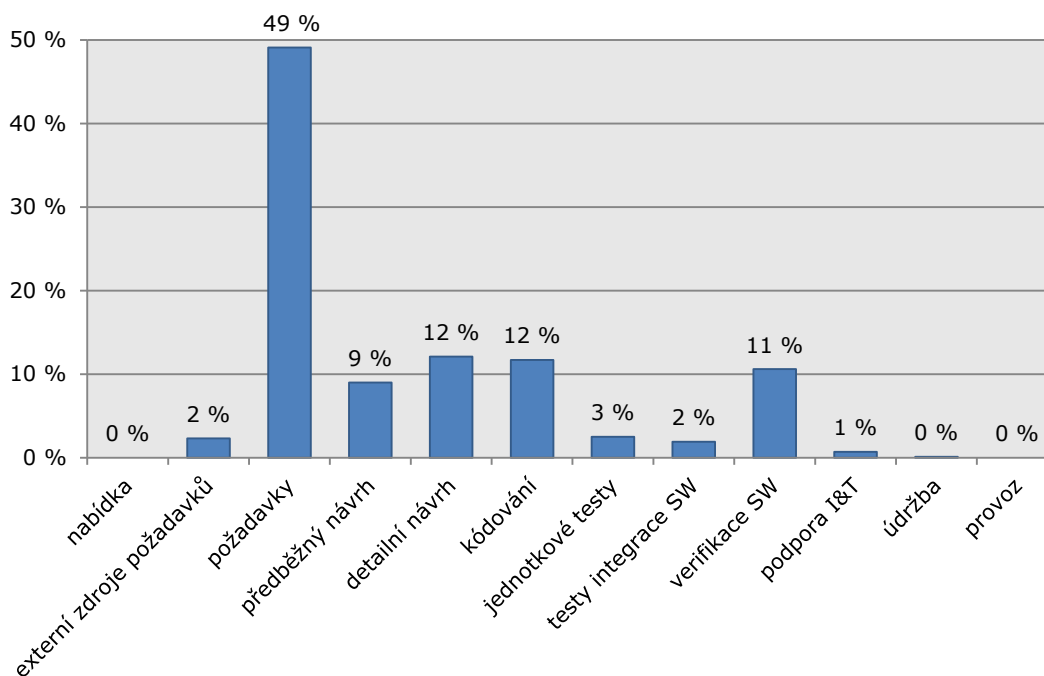
Podle Standish Group jsou hlavními příčinami úspěšnosti projektů IT

- *Zapojení uživatelů*
- Podpora vrcholového vedení
- Jasně a srozumitelné byznys cíle
- Zkušený vedoucí projektu
- Milníky menšího rozsahu
- *Ustálené základní požadavky*

Experti Standish Group zdůrazňují důležitost zapojení uživatelů a dobrého řízení a analýzy jejich požadavků.

Verze 1.2

Obrázek 2, který představuje data jedné skutečné společnosti¹, ukazuje, že téměř 50 % chyb je zapříčiněno etapou požadavků.



Obrázek 2 Zdroje chyb softwaru

¹ Selby, P., Selby, R.W. Measurement-Driven Systems Engineering Using Six Sigma Techniques to Improve Software Defect Detection, Proceedings of 17th International Symposium, INCOSE, June 2007, San Diego.

2. Pojmy a definice

Definice v této části jsou rozděleny do dvou skupin. První charakterizuje pojmy použité ve všech implementačních balíčcích, tj. obecné pojmy, druhá pak pojmy použité v tomto implementačním balíčku, tj. specifické pojmy.

Obecné pojmy

Proces: soubor vzájemně souvisejících nebo vzájemně se ovlivňujících činností, které přetvářejí vstupy na výstupy. [ISO/IEC 12207]

Činnost: soubor souvisejících úkolů procesu. [ISO/IEC 12207]

Úkol: požadovaná, doporučená nebo přípustná akce, jejímž účelem je přispět k dosažení jednoho či více výstupů procesu. [ISO/IEC 12207]

Podúkol: je-li úkol příliš složitý, je rozdělen do podúkolů.

Krok: v implementačním balíčku je úkol rozložen na sled kroků.

Role: definovaná funkce, která je prováděna členem projektového týmu, například testování, evidence, kontrola nebo programování. [ISO/IEC 24765]

Produkt: informace nebo výstup, které mohou být výsledkem (ne nutně) jednoho nebo několika úkolů (například dokumentace návrhu, zdrojový kód).

Artefakt: informace neuvedené v části 5 standardu ISO/IEC 29110, které však mohou být VSE nápomocné při realizaci projektu.

Specifické pojmy

Požadavek: **1.** prohlášení, které určuje, co produkt nebo proces musí splnit, aby poskytoval požadované chování a/nebo výstupy. *IEEE 1220-2005 IEEE Standard for the Application and Management of the Systems Engineering Process*, 3.1.16. **2.** požadavek na systém nebo software, který specifikuje funkci, již systém/softwareový systém nebo komponenta systému/softwareu musí být způsobilé vykonávat. *ISO/IEC 24765, Systems and Software Engineering Vocabulary*. **3.** požadavek specifikující funkci, kterou systém nebo komponenta systému musí být schopny vykonávat. [ISO/IEC 24765]

Analýza požadavků: Proces studia potřeb uživatelů, kterým se dojde k definici požadavků na systém, hardware nebo software. [ISO/IEC 24765]

Dokument požadavků: dokument obsahující jakoukoli kombinaci doporučení, požadavků nebo nařízení, která má být splněna softwareovým balíkem. [ISO/IEC 24765]

Etapa požadavků: časové období v životním cyklu softwaru, během kterého jsou formulovány a dokumentovány požadavky na softwareový produkt. [ISO/IEC 24765]

Verze 1.2

Specifikace požadavků na software (SRS): SRS je specifikace pro konkrétní softwarový produkt, program nebo sadu programů, které provádějí konkrétní funkce v určitém prostředí. SRS může být napsána jedním představitelem nebo více představiteli dodavatele, jedním představitelem nebo více představiteli zákazníka, nebo oběma stranami. [IEEE 830-98]

Dokument SRS obsahuje funkční i nefunkční požadavky.

SRS může být realizována v podobě textového dokumentu, ale může být také vedena v databázi nebo tabulce.

Nefunkční požadavky: požadavek na software, který nepopisuje, co bude software vykonávat, ale jakým způsobem to bude vykonávat. *ISO/IEC 24765, Systems and Software Engineering Vocabulary*. Synonymum *omezení návrhu*. Viz také: funkční požadavek.

POZNÁMKA Nefunkčními požadavky jsou například požadavky na výkon softwaru, požadavky na externí rozhraní softwaru, omezení návrhu softwaru a kvalitativní vlastnosti softwaru. Nefunkční požadavky je někdy obtížné testovat, proto jsou obvykle hodnoceny subjektivně. [ISO/IEC 24765]

Prototyp: 1. funkční nebo nefunkční experimentální model systému nebo části systému. *IEEE 1233, 1998 Edition (R2002) IEEE Guide for Developing System Requirements Specifications*, 3.12. **2.** předběžný vzor, podoba nebo instance systému sloužící jako model pro pozdější stádia nebo konečnou a úplnou verzi systému. *ISO/IEC 24765, Systems and Software Engineering Vocabulary*. **3.** model nebo předběžná implementace softwaru vhodná pro hodnocení návrhu systému, výkonu nebo produkčních možností, nebo pro lepší pochopení požadavků na software. *ISO/IEC 15910:1999, Information technology – Software user documentation process*, 4.41. [ISO/IEC 24765]

Sledovatelnost: udržování komponent (prvků), jejichž původ může být určen. [ISO/IEC 24765]

Matice sledovatelnosti: matice, která zaznamenává vztahy mezi dvěma nebo více produkty vývojového procesu. [ISO/IEC 24765]

3. Vztah k ISO/IEC 29110

Tento implementační balíček pokrývá činnosti vztahující se k *analýze požadavků* popsané v technické zprávě ISO/IEC 29110, části 5-1 pro velmi malé podniky (VSEs) – Základní profil [ISO/IEC 29110].

V této části je uveden přehled aplikovatelných procesů, činností, úkolů a rolí *Řízení projektů* (Project Management – PM) a *Implementace softwaru* (Software Implementation – SI), které se přímo vztahují k tomuto tématu.

- **Proces: Implementace softwaru (SI)**
- **Činnost: SI.2 Analýza požadavků na software**
- **Úkoly a role:**

Seznam úkolů	Role ²
SI.2.1 Na základě aktuálního <i>Plánu projektu</i> přidělit členům pracovního týmu úkoly odpovídající rolím členů.	TL, WT
SI.2.2 Dokumentovat nebo aktualizovat <i>Specifikaci požadavků</i> . Určit informační zdroje (zákazník, uživatelé, předchozí systémy, dokumenty a další) a konzultovat s nimi nebo je projednat, aby byly získány nové požadavky. Analyzovat určené požadavky pro stanovení rozsahu a proveditelnosti. Vytvořit nebo aktualizovat <i>Specifikaci požadavků</i> .	AN, CUS
SI.2.3 Verifikace (ověření) <i>Specifikace požadavků</i> Ověřit správnost a testovatelnost <i>Specifikace požadavků</i> a její soudržnost s <i>Popisem produktu</i> . Přezkoumat také, že požadavky jsou kompletní, jednoznačné a že si vzájemně neodporují. Zjištěné závěry se dokumentují ve <i>Výsledcích verifikace</i> a provádějí se opravy, dokud dokument neschválí AN. Pokud jsou nezbytné významné změny, iniciuje se <i>Změnový požadavek</i> .	AN
SI.2.4 Validace <i>Specifikace požadavků</i> Potvrdit, že <i>Specifikace požadavků</i> vyhovuje potřebám a sjednaným očekáváním, včetně použitelnosti uživatelského rozhraní. Zjištěné závěry se dokumentují ve <i>Výsledcích validace</i> a provádějí se opravy, dokud dokument neschválí CUS.	CUS, AN
SI.2.5 Dokumentovat předběžnou verzi <i>Dokumentace pro uživatele softwaru</i> nebo aktualizovat současnou příručku. (volitelně)	AN

² POZNÁMKA PŘEKLADATELE AN – Analytik (*Analyst*), CUS – Zákazník (*Customer*), TL – Technický vedoucí (*Technical Leader*), WT – pracovní tým (*Work Team*).

Podle ISO/IEC TR 29110-5-1-2:2011 se na úkolu SI.2.3 a SI.2.6 podílí i technický vedoucí (TL).

Verze 1.2

Seznam úkolů	Role²
SI.2.6 Verifikace (ověření) <i>Dokumentace pro uživatele softwaru</i> Ověřit soulad <i>Dokumentace pro uživatele softwaru</i> se <i>Specifikační požadavků</i> . Zjištěné závěry se dokumentují ve <i>Výsledcích verifikace</i> a provádějí se opravy, dokud dokument neschválí AN. Pokud jsou nezbytné významné změny, iniciuje se <i>Změnový požadavek</i> . (volitelně)	AN
SI.2.7 Začlenit <i>Specifikaci požadavků</i> a <i>*Dokumentaci pro uživatele softwaru</i> do potvrzené <i>Konfigurace softwaru</i> . *(volitelně)	TL

4. Popis procesů, činností, úkolů, kroků, rolí a produktů

Proces: Implementace softwaru (SI)

Činnost: SI.2 Analýza požadavků na software

Seznam úkolů	Role ²
SI.2.1 Na základě aktuálního <i>Plánu projektu</i> přidělit členům pracovního týmu úkoly odpovídající rolím členů.	TL, WT
SI.2.2 Dokumentovat nebo aktualizovat <i>Specifikaci požadavků</i> .	AN, CUS
SI.2.3 Verifikace (ověření) <i>Specifikace požadavků</i>	AN
SI.2.4 Validace <i>Specifikace požadavků</i>	CUS, AN
SI.2.5 Dokumentovat předběžnou verzi <i>Dokumentace pro uživatele softwaru</i> nebo aktualizovat současnou příručku. (volitelně)	AN
SI.2.6 Verifikace (ověření) <i>Dokumentace pro uživatele softwaru</i>	AN
SI.2.7 Začlenit <i>Specifikaci požadavků</i> a <i>*Dokumentaci pro uživatele softwaru</i> do potvrzené <i>Konfigurace softwaru</i> . *(volitelně)	TL

Úkoly

Určení požadavků

Cíle	Cílem této činnosti je jasně definovat rozsah projektu a určit klíčové požadavky na systém.
Odůvodnění	Jasná definice rozsahu projektu (hranic) a určení klíčových funkcionalit budoucího systému se zákazníkem jsou důležité proto, aby se předešlo problémům, jako jsou opomenuté klíčové funkce nebo plíživé požadavky.
Role	Vedoucí projektu Analytik
Artefakty	Případy užití – scénáře Dokument požadavků
Kroky	<ol style="list-style-type: none"> 1. Shromážděte informace o aplikační doméně (například finance, zdravotnictví) 2. Určete rozsah projektu 3. Určete a zachyťte požadavky 4. Strukturujte požadavky a určete jejich prioritu

Verze 1.2

<p>Popis kroků</p>	<p>Krok 1. Shromážděte informace o aplikační doméně</p> <p>V průběhu tohoto kroku analytik zachycuje klíčové pojmy problémové oblasti zákazníka. Zákazník je analytikovi nápomocen tím, že mu poskytuje všechny informace (existující dokumentaci či vysvětlení), které usnadní porozumění problémové oblasti zákazníka.</p> <p>Klíčové pojmy jsou zaneseny v části <i>Slovník</i> návrhu dokumentu <i>Specifikace požadavků na software</i>.</p> <p>Krok 2. Určete rozsah projektu</p> <p>Softwarový analytik, za pomoci osoby odpovědné za smluvní hlediska projektu (vedoucího prodeje), zřetelně určí hlavní funkcionality, které spadají do rozsahu projektu.</p> <p><i>Tipy:</i> Pro vyjasnění rozdílů v chápání projektu mezi vámi a zákazníky je také velmi cenné určovat funkcionality, které jsou MIMO rozsah projektu.</p> <p>Krok 3. Určete a zachyťte požadavky</p> <p>Se zřetelem ke klíčovým pojmům vztahujícím se k problémové oblasti zákazníka může analytik začít určovat požadavky. Okolnosti IT projektů nikdy nejsou stejné. V některých případech je většina požadavků již určena v nějakém dokumentu (výzva pro podávání nabídek do výběrového řízení v případě projektů s pevnou cenou), avšak ve většině případů jsou klíčové požadavky zákazníkem pouze (ústně) zmíněny.</p> <p>Analytik musí určit a sepsat klíčové požadavky na systém, který bude vybudován. V průběhu tohoto kroku by analytik neměl určené požadavky začít upřesňovat. Hlavním cílem je získat komplexní pohled na požadavky, které jsou na systém kladeny.</p> <p>Krok 4. Strukturujte požadavky a určete jejich prioritu</p> <p>Analytik musí požadavky určené v předchozím kroku odpovídajícím způsobem uspořádat a strukturovat je (například podle byznys procesů nebo funkcí systému).</p> <p>Priorita klíčových funkcionalit systému musí být určena zákazníkem. Priority mohou být stanoveny jako</p> <ul style="list-style-type: none"> • „Vysoká“ – funkcionalita, která musí být implementována • „Střední“ – funkcionalita, která by měla být implementována • „Nízká“ – funkcionalita, která by mohla být implementována <p>Výstupem tohoto kroku je seznam požadavků, které jsou uspořádány v <i>Dokumentu požadavků</i>.</p>
---------------------------	--

Verze 1.2

Upřesnění a analýza požadavků

Upřesnění a analýza požadavků	
Cíle	Cílem tohoto kroku je podrobný popis a analýza všech určených požadavků.
Odůvodnění	Určené požadavky je důležité projít a odhalit ty požadavky, které se zdají být snadno implementovatelné, ale skrývají složité provádění operací (<i>business complexity</i>), které způsobí problémy při realizaci projektu.
Role	Analytik
	Zákazník
	Vývojář
Artefakty	Případy užití – scénáře
	Dokument požadavků
	Prototyp softwaru
Kroky	1. Upřesněte požadavky
	2. Vytvořte prototyp
Popis kroků	<p>Krok 1. Upřesněte požadavky</p> <p>Analytik projde sadu určených požadavků a provede podrobnější analýzu.</p> <p>Některé požadavky mohou složité provádění operací (<i>business complexity</i>) implicitně zahrnovat, což musí být vyjasněno v této fázi – před jakoukoli implementací.</p> <p>Pokud je to třeba (například když musí být vyvinuto grafické uživatelské rozhraní), bude analytik spolupracovat se zástupci zákazníka na tom, aby byly objasněny ergonomické otázky.</p> <p>Tento krok vyústí v novou verzi dokumentu <i>Specifikace požadavků na software</i>.</p> <p>Krok 2. Vytvořte prototyp</p> <p>Vytvoření prototypu může všem účastníkům projektu (tj. na straně zákazníka i vývojového týmu) usnadnit pochopení požadavků. Prototyp může implementovat pouze některé z funkcionalit.</p>

Verze 1.2

Verifikace (ověření) a validace požadavků

Verifikace (ověření) a validace požadavků	
Cíle	Ověřit (verifikovat) požadavky a získat od zákazníka nebo jeho zástupce potvrzení (validaci) požadavků.
Odůvodnění	Za účelem vyvarování se neustálých zásadních změn požadavků je důležité požádat zákazníka o potvrzení (validaci) požadavků.
Role	Analytik
	Zákazník
	Vedoucí projektu
	Vývojář
Artefakty	Dokument požadavků
	Prototyp softwaru
Kroky	1. Objasněte nejasné požadavky (verifikace)
	2. Přezkoumejte specifikaci požadavků na software
	3. Validujte požadavky
Popis kroků	<p>Krok 1. Objasněte nejasné požadavky</p> <p>Přezkoumejte požadavky, abyste odhalili ty, které nejsou dostatečně jasné (zákazník nebo vývojář softwaru je mohou pochopit rozdílně). Při provádění přezkoumání mohou být použita tato kritéria</p> <ul style="list-style-type: none"> • Zřetelnost (vyvarování se nejednoznačným požadavkům) • Jedinečnost (tj. vyvarování se dvou nebo více požadavků, které vyjadřují stejnou věc) • Proveditelnost (v souladu s přidělenými zdroji projektu) • Testovatelnost <p>Krok 2. Přezkoumejte specifikaci požadavků na software</p> <p>V průběhu tohoto kroku jsou požadavky zevrubně přezkoumávány se zákazníkem za účelem ujištění se, že jsou</p> <ul style="list-style-type: none"> • Kompletní • Správné <p><i>Tipy:</i> Tento krok může být vykonán iterativně přezkoumáváním dané podmnožiny požadavků. Musí být zapojeni softwaroví inženýři, aby identifikovali technické závislosti mezi požadavky (tj. požadavek A musí být z realizačních důvodů implementován před požadavkem B).</p> <p>Krok 3. Validujte požadavky</p> <p>Získejte od svého zákazníka souhlas s požadavky (nebo s danou podmnožinou požadavků, pokud uplatňujete iterativní životní cyklus vývoje softwaru).</p>

Verze 1.2

Řízení změn požadavků

Řízení změn požadavků	
Cíle	Řídit změny požadavků podle procesu, který byl předem smluven se zákazníkem.
Odůvodnění	Změna požadavků je trvalým prvkem většiny IT projektů. Řízení změn v projektu musí být plánováno a předem smluveno se zákazníkem.
Role	Analytik
	Vedoucí projektu
	Zákazník
Artefakty	Dokument požadavků
Kroky	1. Sledujte změny požadavků
	2. Analyzujte dopad změn
	3. Určete změny, které jsou mimo rozsah projektu
	4. Určete priority změn
Popis kroků	<p>Krok 1. Sledujte změny požadavků</p> <p>Tento krok se zaměřuje na sběr a řízení jakýchkoli změn požadavků v centrálním úložišti (může jít o tabulku nebo jakoukoli databázi) vyjádřených oproti existujícím požadavkům. Sledování změn požadavků zahrnuje změny stávajících požadavků, ale také nové nebo odstraněné požadavky.</p> <p>Krok 2. Analyzujte dopad změn</p> <p>Určete dopad každé požadované změny na časový plán a náklady projektu.</p> <p>Krok 3. Určete změny, které jsou mimo rozsah projektu</p> <p>Analytik, za pomoci osoby odpovědné za smluvní hlediska projektu (vedoucího prodeje), určí změny, které jsou mimo rozsah projektu. Změny, které mohou ovlivnit rozpočet projektu, by měly být projednány se zákazníkem.</p> <p>Krok 4. Určete priority změn</p> <p>V průběhu tohoto kroku vedoucí projektu musí za účelem úpravy plánování projektu získat od zákazníka priority určených změn.</p>

Verze 1.2

Role

Role	Definice
Analytik (Analyst)	Osoba v rámci vývojového týmu, která je odpovědná za sběr, analýzu a řízení požadavků vztahujících se k softwaru, jenž má být vyvinut.
Zákazník (Customer)	Osoba na straně zadavatele, která je odpovědná za postoupení požadavků vývojovému týmu a jejich validaci. Může se jednat o zákazníka nebo jakéhokoli zástupce.
Vývojář (Developer)	Osoba odpovědná za vývoj softwaru.
Vedoucí projektu (Project Manager)	Osoba odpovědná za řízení projektu (náklady, časový plán, úkoly, smlouvy, ...).

Tabulka 1 Definice rolí**Artefakty**

Role	Definice
Případy užití – scénáře	Popis sledu interakcí mezi uživatelem a budoucími systémy. Případy užití mohou být zaznamenány tak, jak je předepsáno UML, ale mohou být vyjádřeny také textovými scénáři.
Dokument požadavků ³	Dokument, ve kterém jsou soustředěny všechny určené požadavky, viz <i>Specifikace požadavků na software</i> v části 2.
Prototyp softwaru	Pracovní software vytvořený během časných fází za účelem předvedení/validace funkcionality systému.

Tabulka 2 Definice artefaktů

³ POZNÁMKA PŘEKLADATELE V českém jazyce zpravidla hovoříme o *specifikaci požadavků*. Pojem *specifikace požadavků* by však nedovolil zachovat odlišnosti originálu (*Requirements Document* a *Software Requirements Specification*) a pojem *dokumentace požadavků* by mohl působit jako soubor mnoha dokumentů různé povahy. Pro překlad tohoto implementačního balíčku bylo proto zavedeno označení artefaktu *dokument požadavků*, ale doporučuje se, aby uživatelé balíčku zvolili takové označení dokumentu, které je pro ně dostatečně přesné i srozumitelné a které jim usnadní užívání standardu.

Verze 1.2

5. Šablony

Šablony poskytované v tomto implementačním balíčku by měly být přizpůsobeny pro váš projekt.

Šablona Obsahu SRS – Základní seznam požadavků

Tato šablona je určena pro použití v tabulce strukturované – například – takto

<i>ID</i>	<i>Požadavek</i>	<i>Popis</i>	<i>Priorita</i>

Šablona Obsahu SRS – přizpůsobeno podle IEEE 830

1. Úvod

- 1.1 Účel
- 1.2 Konvence dokumentu
- 1.3 Zamýšlení příjemci dokumentu
- 1.4 Doplnující informace
- 1.5 Kontaktní informace / členové týmu SRS
- 1.6 Odkazy

2. Globální popis

- 2.1 Kontext produktu
- 2.2 Funkce produktu
- 2.3 Třídy a charakteristika uživatelů
- 2.4 Provozní prostředí
- 2.5 Uživatelské prostředí
- 2.6 Omezení návrhu a implementace
- 2.7 Předpoklady a závislosti

3. Požadavky na vnější rozhraní

- 3.1 Uživatelská rozhraní
- 3.2 Hardwarová rozhraní
- 3.3 Softwarová rozhraní
- 3.4 Komunikační protokoly a rozhraní

4. Vlastnosti systému

- 4.1 Vlastnost A systému
 - 4.1.1 Popis a priorita
 - 4.1.2 Akce/výsledek
 - 4.1.3 Funkční požadavky
- 4.2 Vlastnost B systému

...

Verze 1.2

5. Jiné nefunkční požadavky

5.1 Požadavky na výkon

5.2 Požadavky na bezpečnost

5.3 Požadavky na zabezpečení

5.4 Kvalitativní znaky softwaru

5.5 Projektová dokumentace

5.6 Uživatelská dokumentace

6. Jiné požadavky

Příloha A: Slovník / seznam pojmů a definic

Příloha B: K dořešení (*To be determined*)

Verze 1.2

Šablona Obsahu SRS – Construx⁴

- 1 Úvod**
 - 1.1 ÚČEL
 - 1.2 DEFINICE, ZKRATKY A ACRONYMY
 - 1.3 ODKAZY
 - 1.4 PŘEHLED
- 2 Celkový popis**
 - 2.1 KONTEXT PRODUKTU
 - 2.1.1 *Rozhraní systému*
 - 2.1.2 *Uživatelská rozhraní*
 - 2.1.3 *Hardwarová rozhraní*
 - 2.1.4 *Softwarová rozhraní*
 - 2.1.5 *Komunikační rozhraní*
 - 2.1.6 *Paměťová omezení*
 - 2.1.7 *Provoz*
 - 2.1.8 *Požadavky na přizpůsobení okolí*
 - 2.2 FUNKCE PRODUKTU
 - 2.3 CHARAKTERISTIKY UŽIVATELŮ
 - 2.4 OMEZENÍ
 - 2.5 PŘEDPOKLADY A ZÁVISLOSTI
 - 2.6 ROZVRŽENÍ POŽADAVKŮ⁵
- 3 Vlastní požadavky**
 - 3.1 POŽADAVKY VNĚJŠÍCH ROZHRAŇÍ
 - 3.1.1 *Uživatelská rozhraní*
 - 3.1.2 *Hardwarová rozhraní*
 - 3.1.3 *Softwarová rozhraní*
 - 3.1.4 *Komunikační rozhraní*
 - 3.2 VLASTNOSTI SOFTWARE PRODUKTU
 - 3.2.1 *Vlastnost 1*
 - Účel
 - Sled podnětu a odezvy
 - Související funkční požadavky
 - 3.3 POŽADAVKY NA VÝKON
 - 3.4 OMEZENÍ NÁVRHU
 - 3.5 VLASTNOSTI SOFTWARE SYSTÉMU
 - 3.5.1 *Spolehlivost*
 - 3.5.2 *Dostupnost*
 - 3.5.3 *Zabezpečení*
 - 3.5.4 *Udržovatelnost*
 - 3.6 POŽADAVKY NA DATABÁZE [LOGICKÁ ÚROVEŇ]
 - 3.7 OSTATNÍ POŽADAVKY

⁴ www.construx.com

⁵ POZNÁMKA PŘEKLADATELE Tato část specifikace zpravidla určuje, které požadavky mohou být odloženy do dalších verzí produktu. Tvůrce specifikace zde může například vymezit, jakou prioritu mají požadavky, které musí být implementovány v konkrétní verzi produktu pro dodání, případně rozdělit požadavky mezi jednotlivé plánované vývojové verze v iterativním modelu vývoje softwaru. V angličtině se tato část specifikace označuje pojmem *Apportioning of Requirements*.

Verze 1.2

Šablona Obsahu SRS – Volere⁶**Obsah⁷****Účel a hlavní strany projektu (*Project Drivers*)**

1. Účel projektu
2. Klient, zákazník a další zainteresované strany
3. Uživatelé produktu

Omezení projektu

4. Závazná omezení
5. Jmenné konvence a definice
6. Významné skutečnosti a předpoklady

Funkční požadavky

7. Rozsah práce
8. Rozsah produktu
9. Funkční a datové požadavky

Nefunkční požadavky

10. Požadavky na vzhled a ovládání
11. Požadavky na použitelnost a lidská hlediska
12. Požadavky na výkon
13. Požadavky na provoz a prostředí
14. Požadavky na udržitelnost a podporu
15. Požadavky na zabezpečení
16. Kulturní a politické požadavky
17. Právní a zákonné požadavky

Projektové otázky a problémy

18. Řešené otázky a problémy
19. Uplatněná standardní řešení
20. Nové problémy
21. Úkoly
22. Přejít na nový produkt (migrace)
23. Rizika
24. Náklady
25. Uživatelská dokumentace a školení
26. Čekající a odložené požadavky („čekárna“)⁸
27. Návrhy řešení⁹

⁶ <http://atlsysguild.com/GuildSite/Robb/Template.html>

⁷ POZNÁMKA PŘEKLADATELE Aktualizovaný obsah této šablony s podrobnějším členěním částí a některými vysvětleními i příklady je k nahlédnutí na <http://www.volere.co.uk/template.htm>

⁸ POZNÁMKA PŘEKLADATELE Účelem této části specifikace je zachytit nápady a možné budoucí požadavky, které z různých důvodů nebudou realizovány v následujícím vydání produktu. Samostatné zachycení takových nápadů a požadavků umožňuje lépe využít kreativity zákazníka i uživatelů, neboť záznamem návrhů uživatelů a zákazníka tvůrce softwaru dává najevo, že tyto návrhy považuje za důležité, a zřetelné oddělení od ostatních požadavků tvůrce softwaru současně umožňuje lépe řídit očekávání týkající se právě připravované verze produktu.

⁹ POZNÁMKA PŘEKLADATELE Ve fázi specifikace požadavků je třeba určit reálné potřeby a požadavky, nikoli navrhnout technická nebo jiná řešení, ale lidé mají sklon při specifikování požadavků různá možná řešení zmiňovat nebo vymýšlet. Účelem této části specifikace je oddělit návrhy řešení od požadavků a současně tyto návrhy zaznamenat, neboť mohou být později užitečné.


Verze 1.2

ČÍSLO POŽADAVKU Jedinečný identifikátor	TYP POŽADAVKU Typ z předlohy	ČÍSLA UDÁLOSTÍ/PŘÍPADŮ UŽITÍ Seznam událostí / případů užití, které vyžadují tento požadavek
POPIS	Záměr požadavku vyjádřený jednou větou	
ZDŮVODNĚNÍ	Odůvodnění požadavku	
PŮVODCE	Osoba, která vznesla požadavek	
KRITÉRIA SPLNĚNÍ POŽADAVKU		
Takový způsob ověření požadavku nebo ukazatel a jeho hodnota, kterými je možné otestovat, zda řešení odpovídá původnímu požadavku		
SPOKOJENOST ZÁKAZNÍKA		NESPOKOJENOST ZÁKAZNÍKA
PRIORITA	Klasifikace hodnoty pro zákazníka	KONFLIKTY
PODPŮRNÉ MATERIÁLY	Odkaz na dokumenty, které ukazují a vysvětlují tento požadavek	
HISTORIE	Vytvoření, změny, odstranění aj.	

Míra spokojenosti zákazníka spojená s úspěšnou implementací požadavku, v rozsahu 1 (nezúčastněný) až 5 (extrémně potěšený).

Míra nespokojenosti zákazníka, pokud požadavek nebude součástí konečného produktu, v rozsahu 1 (sotva to má význam) až 5 (extrémně nespokojený).

Jiné požadavky, které nemohou být implementovány, pokud je implementován tento požadavek.



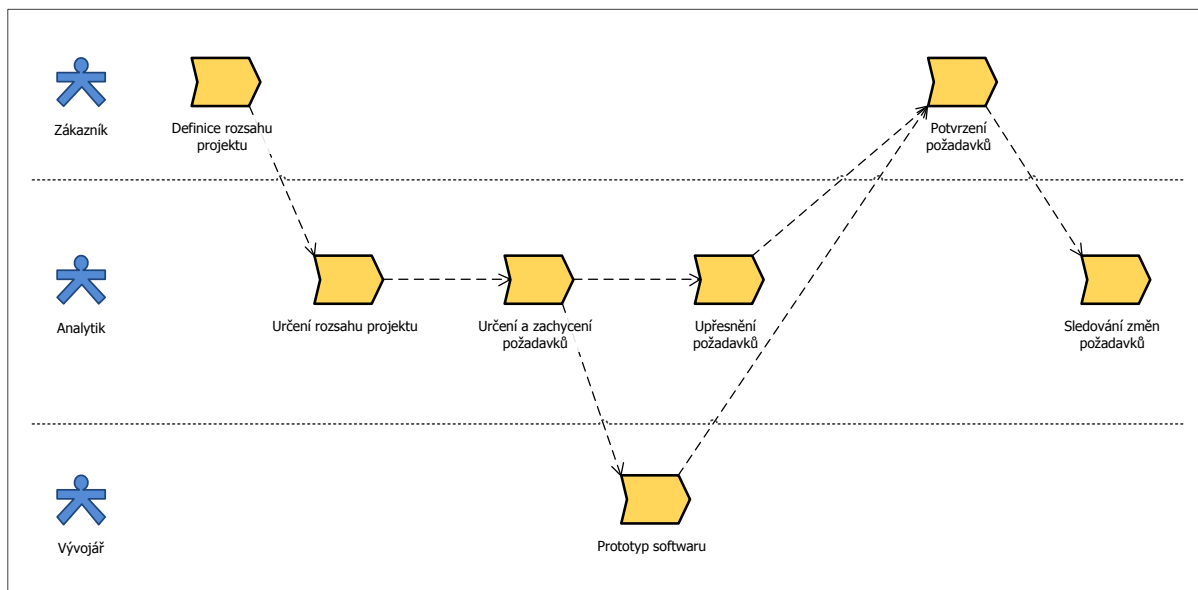
COPYRIGHT © ATLANTIC SYSTEMS GUILD

Šablona obsahu SRS – Karta požadavku – Volere

6. Příklady postupu činností

Upozornění: Tato část obsahuje grafická znázornění příkladů postupu činností analýzy požadavků na software. Tyto příklady jsou poskytovány jako pomoc čtenáři při implementaci jeho vlastního postupu činností analýzy požadavků, který bude odpovídat kontextu a omezením jeho IT projektu.

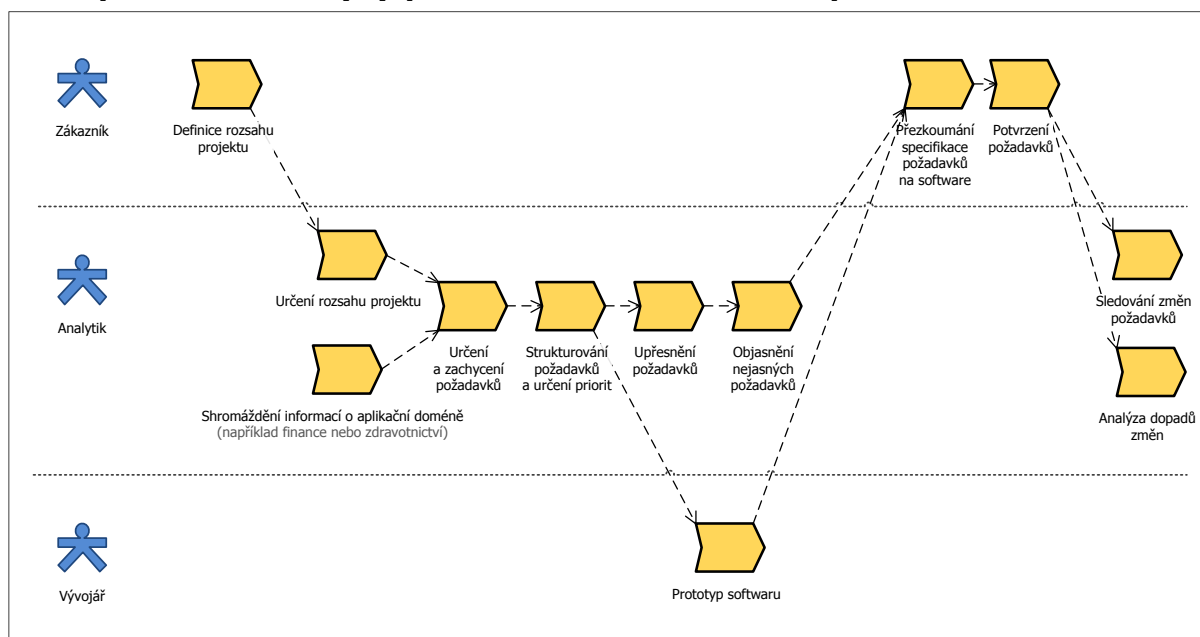
Postup činností analýzy požadavků na software – příklad 1



Obrázek 3 Postup činností analýzy požadavků na software – příklad 1

Verze 1.2

Postup činností analýzy požadavků na software – příklad 2



Obrázek 4 Postup činností analýzy požadavků na software – příklad 2

7. Kontrolní seznam

Kontrolní seznam požadavků

Tento kontrolní seznam pro ověření požadavků vychází z [Constr07]

RS 1 Testovatelnost	Všechny požadavky jsou objektivně ověřitelné.
RS 2 Úplnost	Jsou požadavky kompletní?
RS 3 Sledovatelnost	Všechny požadavky musí být sledovatelné až ke specifikaci systému, článku smlouvy nebo nabídky.
RS 4 Správnost	Požadavky jsou správné – tj. přesně odrážejí požadavky zákazníka.
RS 5 Jedinečnost	Tentýž požadavek musí být stanoven pouze jednou.
RS 6 Jednoduchost	Požadavky musí být rozloženy do jejich nejprostší podoby.
RS 7 Rozsah¹⁰	Spadají požadavky do rozsahu?
RS 8 Vysoká úroveň popisu	Požadavek musí být stanoven jako nejvyšší potřeba, nikoli jako vnímané prostředky řešení.
RS 9 Kvalita	Kvalitativní znaky jsou definovány.
RS 10 Jednoznačnost	Specifikace musí obsahovat vyjádření požadavků, která lze interpretovat jen jedním způsobem. ¹⁰
RS 11 Hardware	Hardwarové prostředí je zcela přesně popsáno.
RS 12 Stálost	Požadavky jsou pevným základem pro návrh.

¹⁰ POZNÁMKA PŘEKLADATELE V anglickém znění implementačního balíčku jsou RS 7 a popisek RS 10 označeny jako netisknutelný text, který se nezobrazuje ani netiskne. V tomto překladu jsou oba texty zobrazeny.

Verze 1.2

8. Nástroje

Nástroj pro sledovatelnost

- Cíle
 - Udržet vazbu ze zdroje každého požadavku přes jeho dekompozici, implementaci až k testování (verifikaci).
 - Zajistit, že všechny požadavky jsou vypořádány a že je vyvíjeno pouze to, co je požadováno.
 - Napomoci při provádění posouzení dopadu změn požadavků, návrhu nebo jiných změn přizpůsobených položek.

Matice sledovatelnosti									
Datum (RRRR-MM-DD): _____									
Název projektu: _____									
	Jméno			Podpis			Datum (RRRR-MM-DD)		
Ověřil:	_____			_____			_____		
Schválil:	_____			_____			_____		
Identifikační číslo	Text potřeby	Text požadavku	Způsob ověření	Název nebo ID případu užití	Název nebo ID modulu kódu	Název nebo ID testovací procedury	Datum ověření	Jméno osoby, která provedla ověření	Výsledek ověření

Návod k užití	
<p>Výše uvedená tabulka by měla být vytvořena v tabulkovém procesoru nebo databázi tak, aby mohla být řazena podle kteréhokoli sloupce a umožnila obousměrnou sledovatelnost mezi sloupci. Jedinečné identifikátory položek by měly být přiřazeny v podobě hierarchické osnovy, v níž pro položky nižší úrovně (tj. podrobnější) mohou být zjištěny položky úrovně vyšších.</p>	
Jedinečná identifikace požadavku (ID)	Jedinečné ID požadavku / prohlášení o požadavcích na systém, ve kterém je požadavek zmíněn, a/nebo jedinečná identifikace pro požadavky vzniklé dekompozicí.
Popis požadavku	Zadejte popis požadavku (například popis žádosti o změnu – <i>Change Request</i>).
Odkaz na návrh	Zadejte číslo odstavce, ve kterém je CR uveden v dokumentaci návrhu.
Odkaz na modul / konfigurovanou položku	Zadejte jedinečný identifikátor softwarového modulu nebo konfigurované položky, které realizují návrh.
Odkaz na vydání (release)	Zadejte číslo verze vydání / buildu, ve které je požadavek naplněn.
Odkaz na název / číslo kroku testovacího skriptu	Zadejte název / číslo kroku testovacího skriptu, ve kterém je požadavek odkazován (například <i>Krok 1</i>).

Verze 1.2

Pokyny

Sledovatelnost požadavků by měla

- zajistit sledovatelnost pro každou úroveň dekompozice provedenou na projektu, zejména
 - zajistit, že pro každý požadavek nižší úrovně může být zjištěn požadavek vyšší úrovně nebo původní zdroj
 - zajistit, že pro každý prvek návrhu, implementace a testování může být zjištěn výchozí požadavek
 - zajistit, že každý požadavek je ztvárněn v návrhu a implementaci
 - zajistit, že každý požadavek je zastoupen při testování/verifikaci
- zajistit, že sledovatelnost je využívána při provádění posouzení dopadu změn požadavků na plány projektu, činnosti a pracovní produkty
- být udržována a aktualizována, jakmile se vyskytne změna
- být brána v úvahu během přípravy *Posouzení dopadu* pro každou navrhovanou změnu projektu
- být naplánována, protože udržování vazeb/odkazů je pracovně namáhavý proces, který by měl být zaznamenáván/monitorován a měl by být přiřazen členovi projektového týmu
- být udržována jako elektronický dokument

9. Odkazy na jiné standardy a modely

Tato část uvádí odkazy tohoto implementačního balíčku na standardy ISO/IEC a Capability Maturity Model IntegrationSM verze 1.2 od Software Engineering Institute (CMMI[®]).¹¹

Poznámky

- Tato část je poskytována pouze pro informaci.
- Každá tabulka uvádí pouze úkoly, které jsou pokryty tímto implementačním balíčkem.
- Tabulky užívají následující konvenci
 - plné pokrytí = F (Full Coverage)
 - částečné pokrytí = P (Partial Coverage)
 - nepokryto = N (No Coverage)

Poznámka: Matice pokrytí nejsou kompletní, jsou poskytovány pouze jako příklad, který ilustruje, jak matice dokončit.

Matice odkazů k ISO 9001

Název úkolu a kroku	Pokrytí F/P	Článek ISO 9001	Poznámky
Určení požadavků Krok 1 – Shromážděte informace o aplikační doméně	F	7.2.1 Určování požadavků týkajících se produktu [Organizace musí určit] b) požadavky, které zákazník nevedl, ale které jsou nezbytné pro specifikované nebo zamýšlené použití, je-li toto použití známo	
Určení požadavků Krok 2 – Určete rozsah projektu	F	7.2.1 Určování požadavků týkajících se produktu Organizace musí určit a) požadavky specifikované zákazníkem, včetně požadavků na činnosti při dodání a po dodání	

¹¹ SM CMM Integration je značka služby Carnegie Mellon University.

[®] Capability Maturity Model a CMMI jsou registrované známky Carnegie Mellon University, které jsou zapsány u U.S. Patent and Trademark Office.

Verze 1.2

Matice odkazů k ISO/IEC 12207

Název úkolu a kroku	Pokrytí F/P	Článek ISO/IEC 12207	Poznámky
<i>Určení požadavků</i> Krok 1 – Shromážděte informace o aplikační doméně			
<i>Určení požadavků</i> Krok 2 – Určete rozsah projektu	P	7.1.2 Proces analýzy požadavků na software 7.1.2.2 Výsledky	

Matice odkazů pro CMMI

Název úkolu a kroku	Pokrytí F/P	Cíl / Praktika CMMI V1.2	Poznámky
<i>Určení požadavků</i> Krok 1 – Shromážděte informace o aplikační doméně	P		
<i>Určení požadavků</i> Krok 2 – Určete rozsah projektu	P	SG 1 Rozpracujte požadavky zákazníka	

Verze 1.2

10. Odkazy

Zkratka odkazu	Odkaz
[ISO/IEC 29110]	ISO/IEC TR 29110-5-1-2:2011, Software Engineering – Lifecycle Profiles for Very Small Entities (VSEs) – Part 5-1: Management and Engineering Guide – Basic VSE Profile. ¹²
[OWPL-EN]	Renault A., Habra N., Alexandre S., Deprez J.-C., OWPL. Software Process Improvement for VSE, SME and low maturity enterprises. Version 1.2.2, FUNDP-CETIC, 2000. (http://www.cetic.be/internal393.html)
[IEEE 830-98]	IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE, 1998.
[ISO/IEC 12119]	ISO/IEC 12119:1994, Information technology – Software packages – Quality requirements and testing.
[ISO/IEC 12207]	ISO/IEC 12207:2008, Systems and software engineering – Software life cycle processes.
[ISO/IEC 24765]	ISO/IEC 24765, Systems and Software Engineering Vocabulary
[ConstSoft02]	Construx Software – Checklist for Software Requirements Specifications, 2002.
[SELB07]	Selby, P., Selby, R.W., Measurement-Driven Systems Engineering Using Six Sigma Techniques to Improve Software Defect Detection, Proceedings of 17 th International Symposium, INCOSE, June 2007, San Diego.
[STAN02]	Standish Group – Chaos report 2002.
[SPEM05]	Software Process Engineering Metamodel Specification, OMG, 2005.
[VOLE07]	Volere, Requirements Resources – http://www.volere.co.uk

¹² POZNÁMKA PŘEKLADATELE Volně dostupný standard (technická zpráva) ISO/IEC TR 29110-5-1-2:2011 je k dispozici ke stažení na <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>

Verze 1.2

11. Hodnoticí formulář

Formulář se vztahuje k původní anglické verzi dokumentu a vyplňuje se v anglickém jazyce.

<p align="center">Deployment Package – Software Requirements Analysis – Version 1.2</p> <p>Your feedback will allow us to improve this package, your comments and suggestions are welcomed.</p>
<p>1. How satisfied are you with the CONTENT of this deployment package?</p> <p><input type="checkbox"/> <i>Very Satisfied</i> <input type="checkbox"/> <i>Satisfied</i> <input type="checkbox"/> <i>Neither Satisfied nor Dissatisfied</i> <input type="checkbox"/> <i>Dissatisfied</i> <input type="checkbox"/> <i>Very Dissatisfied</i></p>
<p>2. The sequence in which the topics are discussed, is logical and easy to follow?</p> <p><input type="checkbox"/> <i>Very Satisfied</i> <input type="checkbox"/> <i>Satisfied</i> <input type="checkbox"/> <i>Neither Satisfied nor Dissatisfied</i> <input type="checkbox"/> <i>Dissatisfied</i> <input type="checkbox"/> <i>Very Dissatisfied</i></p>
<p>3. How satisfied were you with the APPEARANCE/FORMAT of this deployment package?</p> <p><input type="checkbox"/> <i>Very Satisfied</i> <input type="checkbox"/> <i>Satisfied</i> <input type="checkbox"/> <i>Neither Satisfied nor Dissatisfied</i> <input type="checkbox"/> <i>Dissatisfied</i> <input type="checkbox"/> <i>Very Dissatisfied</i></p>
<p>4. Have any unnecessary topics been included? (please describe)</p>
<p>5. What missing topic would you like to see in this package? (please describe)</p> <ul style="list-style-type: none">• Proposed topic• Rationale for new topic
<p>6. Any error in this deployment package?</p> <ul style="list-style-type: none">• Please indicate<ul style="list-style-type: none">• Description of error• Location of error (section #, figure #, table #)
<p>7. Other feedback or comments</p>
<p>8. Would you recommend this Deployment package to a colleague from another VSE?</p> <p><input type="checkbox"/> <i>Definitely</i> <input type="checkbox"/> <i>Probably</i> <input type="checkbox"/> <i>Not Sure</i> <input type="checkbox"/> <i>Probably Not</i> <input type="checkbox"/> <i>Definitely Not</i></p>

Optional

- Name _____
- e-mail address _____

Send this form to simon.alexandre@cetic.be or claudio.y.laporte@etsmtl.ca or Avumex2003@yahoo.com.mx