

Implementační balíček

Implementace softwaru

Vstupní profil

Autoři	G. Hernandez École de Technologie Supérieure (ETS), (Canada) W. Gonzalez - École de Technologie Supérieure (ETS), (Canada)
Editor	C. Y. LAPORTE – École de Technologie Supérieure (ETS), (Canada)
Datum vytvoření	25/06/2010
Poslední verze	Sept 21, 2012
Status	Draft
Verze	0.3

Historie verzí

Datum	verze	Autor	Modifikace
25/06/2010	0.1	G. Hernandez W. Gonzalez	Vytvoření dokumentu
19/11/2010	0.2	G. Hernandez W. Gonzalez	Verze
July 9 th, 2012	0.3	C. Laporte	Verze

Zkratky/Označení

Zkr./ozn.	Definice
DP	Implementační balíček (<i>Deployment Package</i>) – soubor artefaktů vyvinutých za účelem usnadnění implementace skupiny praktik vybraného rámce pro velmi malý podnik.
VSE	Velmi malý podnik (<i>Very Small Entity</i>) – podnik, organizace, oddělení nebo projekt do 25 zaměstnanců.
SOW	Specifikace prací (<i>Statement of work</i>)
SI	Softwarová Implementace (<i>Software Implementation</i>)
WBS	Diagram rozkladu prací (<i>Work Breakdown Structure</i>)

1. Technický popis.....	4
<i>Proč je SW implementace důležitá?</i>	4
Definice.....	6
<i>Základní pojmy.....</i>	6
<i>Specifické pojmy.....</i>	6
3. Vztah k normě ISO/IEC 29110.....	8
Popis procesů, činností, úkolů, postupů, rolí a produktů	9
4.1. Činnosti softwarové implementace.....	12
4.1.1. Činnosti: SI.1 Zahájení Softwarové implementace	13
4.1.2. Činnosti: SI.2 Analýza softwarových požadavků	14
4.1.3. Činnosti: SI.3 Identifikace softwarových komponent.....	16
4.1.4. Činnost: SI.4 Tvorba softwaru	18
4.1.5 Činnost: SI.5 Softwarová integrace a testování	21
4.1.6. činnost: SI.6 Dodání produktu	24
4.2 Popis rolí	26
4.3. Popis produktu	26
4.4. Popis artefaktů.....	32
5. Šablony a nástroje.....	33
5.1. Softwarová implementace - vstupní šablona	33
5.2. Šablona - Analýz softwarových požadavků.....	33
5.3 Šablona – Identifikace softwarových komponent	34
5.4 Tvorba softwaru	35
5.5 Šablona - Specifikace testů.....	35
5.6 Šablona – Dodávka produktu	38

6. Příklad činnosti životního cyklu	41
Příklad životního cyklu softwarové implementace	41
7. Kontrolní seznam	43
7.1 Kontrolní seznam Analýzy softwarových požadavků	43
7.2 Kontrolní seznam Identifikace softwarové komponenty	43
7.3 Kontrolní seznam konstrukce software	43
7.4 Kontrolní seznam Softwarové Integrace a Testů	45
7.5 Kontrolní seznam Dodání produktu	45
8. Reference	47
9. Hodnotící formulář	48

1. Technický popis

Účel tohoto dokumentu

Tento implementační balíček (DP) podporuje Vstupní profil jak je definovaný v ISO/IEC TR 29110-5-1-1:2012, Příručka pro řízení a implementaci [ISO/IEC 29110]. Vstupní profil je jedním ze skupiny obecných profilů. Skupina obecných profilů je použita pro velmi malé podniky (skupina lidí do 25 osob, přičemž může jít o podnik, oddělení, projekt či skupinu osob) pracujících na malých projektech (např. Nanejvýš šest měsíců práce člověko-hodin) a pro VSE start-upy. Vstupní profil poskytuje základ pro migraci na procesy Základního profilu.

Implementační balíček je soubor artefaktů vyvinutých za účelem usnadnění implementace skupiny praktik vybraného rámce pro velmi malý podnik (VSE). DP není proces referenčního modelu (tzn. že není normativní). Prvky typického DP jsou: popis procesů, činnosti, úkoly, role a produkty, šablony, kontrolní seznamy, příklady a nástroje.

Obsah tohoto dokumentu je zcela informativní.

Tento dokument je určen pro použití VSE, pro stanovení postupů, pro zavedení procesů vývojových přístupů a metodik, včetně např. agilních, vývojových, přírůstkových, test driven vývoje, atd. Založených na organizaci nebo projektu pro VSE.

ISO/IEC TR 29110-5-1-1:2012 je dostupné zdarma na následující stránce ISO:

<http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>

Proč je SW implementace důležitá?

Implementace je uskutečnění, realizace nebo plnění plánu, metody nebo konstrukce pro provedení něčeho. Implementace jako taková je akce, která musí sledovat předchozí myšlenky aby se něco uskutečnilo.

V kontextu informačních technologií, implementace zahrnuje všechny procesy, spojené se zaváděním nového správně fungujícího softwaru nebo hardwaru, zahrnující instalaci, konfiguraci, provoz, testování a provádění nutných změn. Slovo nasazení je někdy používáno ve stejném kontextu. [crm.com]

Je důležité mít sadu procesů zapadající do systematické analýzy a způsob jak plnit úkoly. Každá aktivita je zaměřená na dosažení cílů stanovených v pracovní specifikaci a hlavním cílem je, dosáhnout konečného výrobku se všemi atributy, které vyžaduje zákazník s vysokou mírou kvality

Význam každé aktivity softwarové implementace je popsán takto:

- "Sw implementation initiation" umožňuje připravit činnosti pro týmovou práci a všechny potřebné nástroje pro dokončení prací.
- Analýza softwarových požadavků - je důležité jasně definovat rozsah projektu (hranice) a identifikovat klíčové funkcionality budoucího systému ve spolupráci se

zákazníkem aby se předešlo problémům, jako zapomenutá důležitá funkcionality nebo požadavek.

- Identifikace softwarových komponent je základním kamenem softwarového projektu. Neschopnost popsat architektonický návrh, který bude zahrnovat veškeré požadavky, je receptem na katastrofu. Zákazník nezaplatí, pokud návrh nereflektuje všechny požadavky.
- Tvorba Softwaru je klíčová pro programátory, kteří budou mít dostatečnou důvěru k systematické přípravě komponent, které mohou být použity pro stavbu komplexních částí.
- The Software Integrace a testování umožňuje provádět různé typy testů a identifikaci problémů, které musí být odstraněny vývojovým týmem. Různé typy testů jsou prováděny v různých etapách projektu.
- Dodávání produktu řídí probíhající aktivity, nemělo by dojít k nepředvídaným situacím a zpoždění pro získání akceptace výstupů. V opačném případě zákazník nezaplatí.

Definice

V této sekci čtenář nalezne dvě sady definic. První sada definuje termíny používané ve všech Implementačních balíčcích tj. Základní pojmy. V druhé sadě, termíny používané v tomto implementačním balíčku tj. Specifické pojmy.

Základní pojmy

Proces: *soubor vzájemně souvisejících nebo vzájemně působících činností, který mění vstupy na výstupy [ISO/IEC 12207]*

Činnost: skupina souvisejících úloh procesu [ISO/IEC 12207].

Úkol: *požadované, doporučené nebo přípustné akce, přispívající k dosažení jedno nebo více výsledků procesu [ISO/IEC 12207]*

Podúkol: Když je úloha složitá, tak je rozdělena na podúlohy.

Krok: *v implementačním balíčku, úloha rozložená do sledu kroků.*

Role: *definované funkce, které má plnit člen projektového týmu, jako je testování, archivace, kontrola, kódování. [ISO/IEC 24765]*

Produkt: *část informací nebo dodávky, které mohou být produktem (není povinné) jedním nebo více úkoly. (např. návrh dokumentu, zdrojový kód).*

Artefakt: *informace, která není uvedena v ISO/IEC 29110 část 5, ale může napomoci VSE v průběhu projektu*

Specifické pojmy

Smlouva: *Definice pojmů a podmínek, za kterých bude uskutečněn pracovní vztah. [ISO/IEC 12207]*

Výchozí stav: *specifikace nebo produkt, který byl prozkoumán a formálně schválen a slouží jako základ pro další vývoj, a který může být změněn pouze formálními postupy pro kontrolu změn. [ISO/IEC 12207]*

Komponenta: Sada funkčních služeb softwaru, která pokud je nasazena, představuje definovanou sadu funkcí a má unikátní rozpoznatelný název.

Defekt: *Problém, který když není opraven může následně způsobit selhání aplikace nebo chybný výsledek.*

Dodání: *1. Uvolnění systému nebo komponenty zákazníkovi nebo budoucímu uživateli.*

Implementační prostředí: *hardware, software a postup potřebný k nasazení.*

Nefunkcionální Požadavek: *softwarový požadavek který nepopisuje co software bude dělat, ale jak to bude dělat. ISO/IEC 24567, Slovní zásoba pro systémy a softwarový inženýring. Konstrukční omezení (Syn. design constraints), nefunkční požadavky. POZNÁMKA: na příklad, požadavky na výkon softwaru, požadavky na vnější softwarové rozhraní, požadavky na omezení softwaru a požadavky na atributy kvality softwaru. Nefunkcionální požadavky někdy obtížné na testování, často jsou hodnoceny subjektivně.*

Produkt: *výsledek procesu. [ISO/IEC 12207]*

Verze: *konkrétní verze konfigurační položky, která je k dispozici pro konkrétní účel (např. testovací verze). [ISO/IEC 12207]*

Requirement: *1. a statement that identifies what a product or process must accomplish to produce required behaviour and/or results. IEEE 1220-2005 IEEE Standard for the Application and Management of the Systems Engineering Process. 3.1.16. 2. a system or software requirement that specifies a function that a system/software system or system/software component must be capable of performing. ISO/IEC 24765, Systems and Software Engineering Vocabulary. 3. a requirement that specifies a function that a system or system component must be able to perform. [ISO/IEC24765]*

Požadavek: *1. Prohlášení, které popisuje, co musí produkt nebo proces splnit, co produkuje, jak se chová a jaké dává výsledky.*

Analýza požadavků: *Proces zjišťování uživatelských potřeb za účelem definice systému, hardwaru, nebo softwarových požadavků. [ISO/IEC 24765]*

Dokumentace požadavků: *dokument obsahující libovolnou kombinaci požadavků doporučení, či nařízení, které je třeba splnit pomocí softwarového balíku.*

Softwarové testovací prostředí: *podporované softwarové testovací prostředí, které splňuje požadavky na testování, v kterém musí být kontrolovány změny a v případě potřeby možný návrat. (Implementační balíček – Softwarové testování – základní profil)*

Diagram rozkladu prací (WBS): *dodavatelsky orientovaný hierarchický rozklad prací, které musí vykonány projektovým týmem za účelem dosažení cílu projektu a vytvořením požadované dodávky. Organizuje a definuje rozsah celého projektu. [PMI 2008]*

3. Vztah k normě ISO/IEC 29110

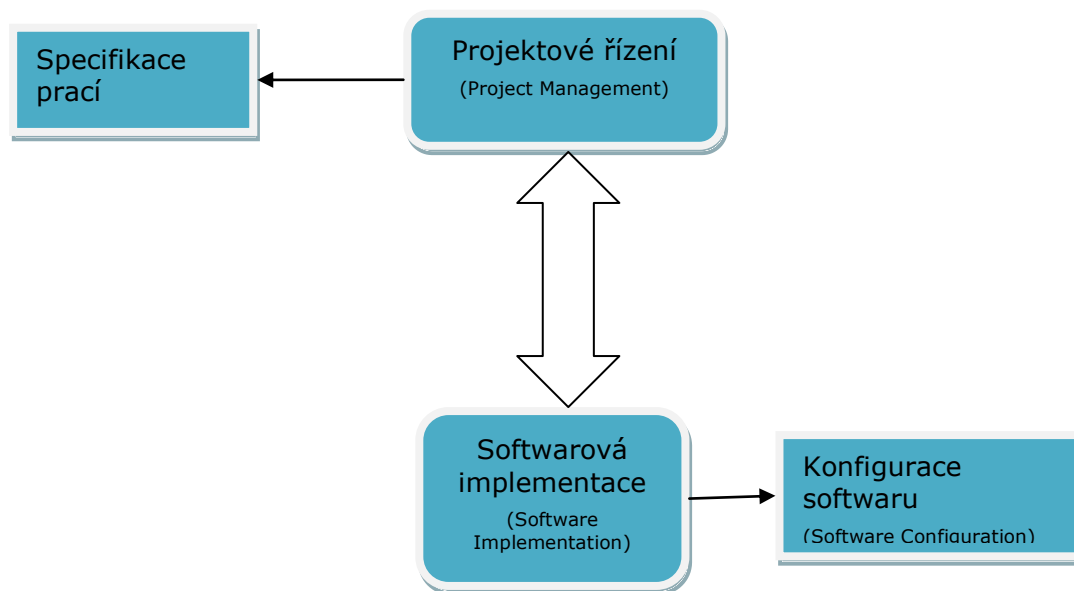
Tento implementační balíček pokrývá aktivity spojené s Projektovým řízením ISO Technical Report ISO/IEC 29110 Part 5-1-1 pro velmi malé podniky (VSEs) – Skupina obecných profilů: Vstupní profil [ISO/IEC 29110, 2011].

Příručka obsahuje Projektové řízení a Softwarovou implementaci procesů, které integruje postupy založené částečně na ISO/IEC 12207:2008- Systémy a softwarové inženýrství – Softwarový životní cyklus procesů a ISO/IEC 15289:2011 Systémy a softwarové inženýrství – Softwarový životní cyklus procesů – pokyny pro obsah procesu životního cyklu informací (dokumentace)

Účelem procesů projektového řízení je stanovit a systematicky provádět úkoly projektu softwarové implementace, v souladu s cíly projektu, v očekávané kvalitě, čase a nákladech.

Účelem procesů softwarové implementace je systematické provádění analýz, návrh, tvorba, integrace a testovací aktivity nových nebo upravených softwarových produktů dle specifických požadavků. Dále implementační balíček popisuje proces projektového řízení podle Vstupního profilu.

Oba procesy jsou vzájemně propojeny (viz obr. 1).



Obrázek 1 – Procesy Vstupního profilu (ISO/IEC 29110)

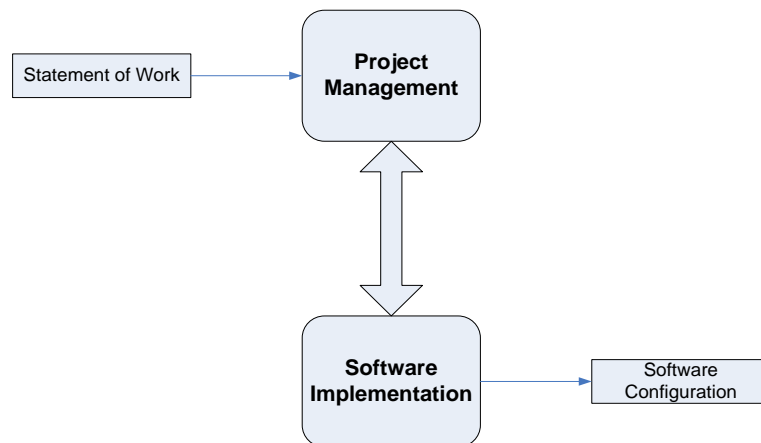
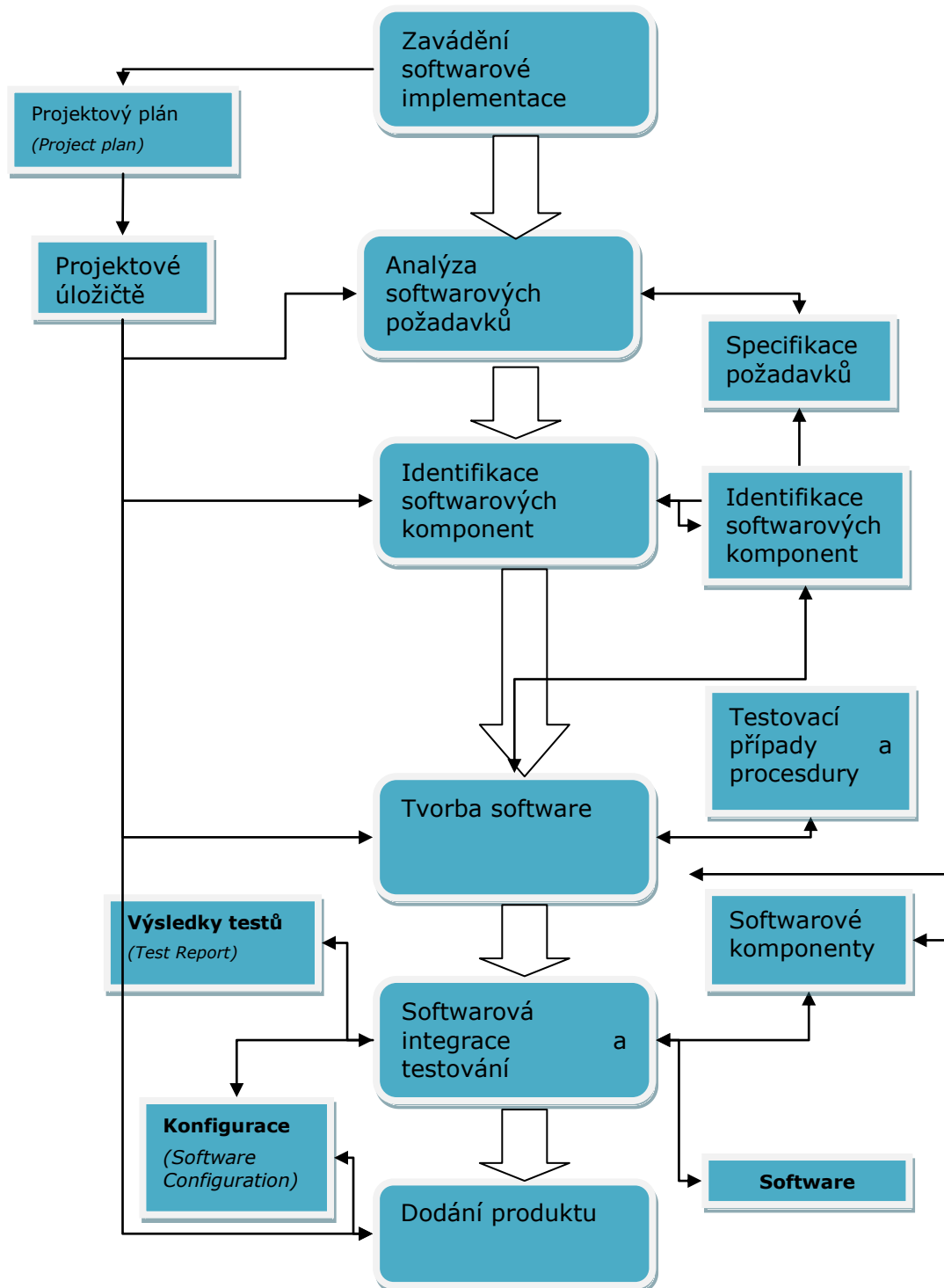


Figure 1 – Entry profile processes (ISO/IEC 29110)

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

Následující diagram zobrazuje tok informací mezi aktivitami procesu softwarové implementace zahrnující nevýznamnější výstupy a jejich vztahy.

Status produktu informuje projektový tým o činnostech, (úkolech) které již byly na projektu dokončeny (hodnocení, schválení, otestování, milníky). Na základě této informace může být zahájena další úloha, kde může být produkt použit jako vstup. Některé produkty nemají přidělený status, protože jsou pouze informativní a nemění svůj obsah (např.: záznam o akceptaci, oprava registru, zálohování projektového úložiště, verifikace/validace výsledků).



Obrázek 2 – Implementace softwaru - process diagram (ISO/IEC 29110)

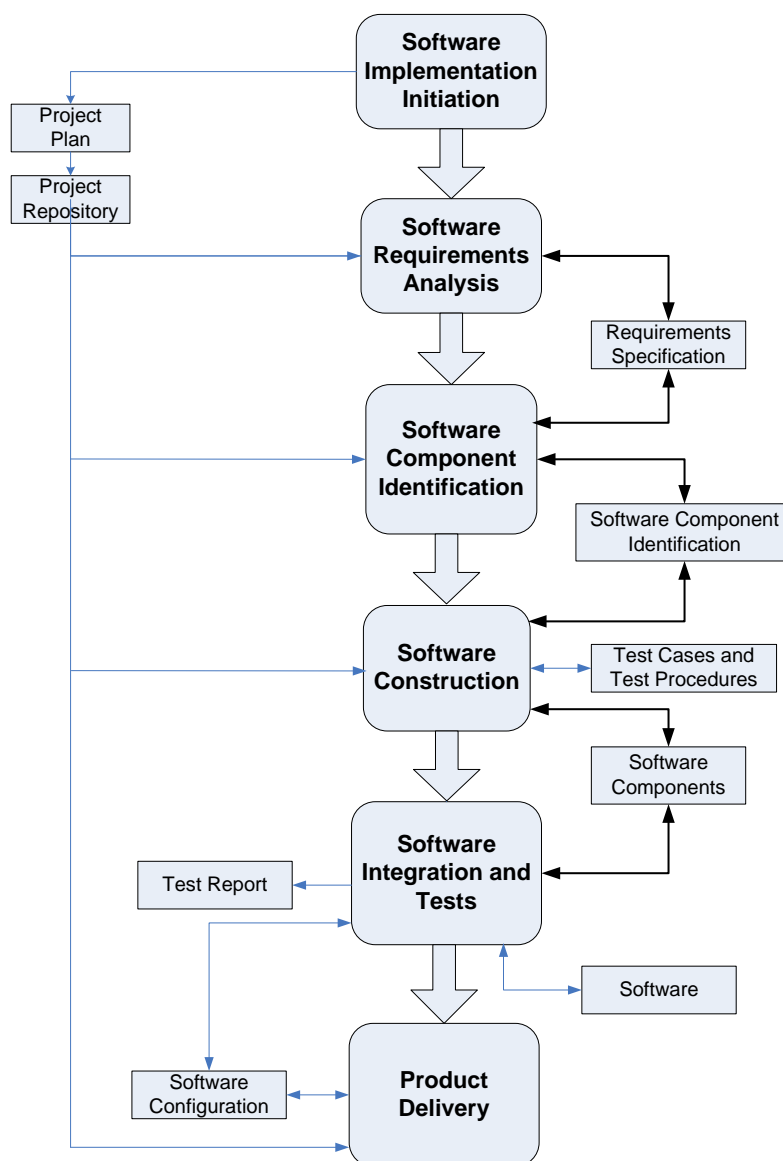


Figure 2 — Software Implementation process diagram (ISO/IEC 29110)

4.1. Činnosti softwarové implementace

Účelem procesu softwarové implementace je systematická analýza, návrh, vytvoření, integrace a testování činností pro nové nebo modifikované softwarové produkty zohledňující specifické požadavky.

Proces softwarové implementace je složen z následujících činností:

© G. Hernandez, W. Gonzalez

1. Zahájení softwarové implementace
2. Analýza požadavků na software
3. Identifikace softwarových komponent
4. Tvorba softwaru
5. Integrace a testování softwaru
6. Nasazení produktu

4.1.1. Činnosti: SI.1 Zahájení Softwarové implementace

Zahájení softwarové implementace zajišťuje, že je projektový tým seznámen s projektovým plánem stanoveným při plánování projektu. Činnosti zajišťují:

- Přezkoumání projektového plánu pracovním týmem za účelem rozdělení úkolů.
- Výběr vývojového prostředí.

Role	Seznam úloh	Vstupní produkt	Výstupní produkt
PM PT (WT)	SI.1.1 Prohlédněte si projektový plán společně s projektovým týmem, tak aby mu porozuměl a zapojil se.	<i>Projektový plán</i> (Project Plan)	<i>Projektový plán [přezkoumaný]</i> (Project Plan [reviewed])
PT (WT)	SI.1.2 Nastavte nebo aktualizujte implementační prostředí.	<i>Projektový plán [přezkoumaný]</i> (Project Plan [reviewed])	

Zahájení Softwarové implementace

Cíle:	Porozumění projektovému plánu všemi členy pracovní skupiny, identifikace a nastavení prvků implementačního prostředí.
Odůvodnění:	To umožňuje připravit pracovní tým na jednotlivé aktivity a poskytne mu nezbytné nástroje pro dokončení práce.
Role:	Project Manager Pracovní tým
Artefakty:	Projektový plán
Postup:	Krok 1: Prostudování projektového plánu pracovním týmem. Krok 2: Nastavení implementačního prostředí.

Popis postupu:	Krok 1: Prostudování projektového plánu pracovním týmem. Projektový manažer by měl mít schůzku za účelem pochopení projektu a cílů všemi zúčastněnými nebo pracovním týmem. Projektový manažer by se měl ujistit, že každý člen týmu byl přizván na tuto schůzku. Na konci schůzky by měl projektový manažer zkontrolovat, že všechny pracovní týmy akceptují a pochopili svou účast na projektu.
	Step 2: Nastavení implementačního prostředí. Během tohoto kroku je nastaveno prostředí, ve kterém bude implementace realizována. Cílem je definovat řízené a nezávislé prostředí pro implementaci.

4.1.2. Činnosti: SI.2 Analýza softwarových požadavků

Činnosti Analýzy softwarových požadavků zkoumají zákazníkovi dohodnuté požadavky a stanoví ověřené požadavky na softwarový projekt. Činnosti zajišťují:

- Pracovní tým přezkoumá projektový plán a vymezí zadání úkolů.
- Získávání, analýza a specifikace zákaznickových požadavků.
- Odsouhlasení zákaznickových požadavků.
- Verifikace a validace požadavků.

Role	Seznam úloh	Vstupní produkt	Výstupní produkt
PM PT (WT)	SI.2.1 Přidělení úkolů členům pracovního týmu podle jejich role, na základě aktuálního projektového plánu.	<i>Project Plan</i> [přepracovaný] (Project Plan [reviewed])	
PT (WT) ZAK (CUS)	SI.2.2 Zdokumentování nebo aktualizace specifikace požadavků. Identifikace a konzultování informačních zdrojů (zákazníci, uživatelé, předchozí systémy, dokumenty, atd.) za účelem získání nových požadavků. Shromáždění a analýza zjištěných požadavků, vymezení rozsahu a proveditelnosti.	<i>Projektový plan</i> (Project Plan) - <i>Popis produktu</i> (Product Description)	<i>Specifikace požadavků</i> [ověřená] (Requirements Specification [verified])

Role	Seznam úloh	Vstupní produkt	Výstupní produkt
	<p>Verifikace správnosti a testovatelnosti specifikovaných požadavků a jejich soulad s popisem produktu.</p> <p>Vytvořit nebo aktualizovat specifikaci požadavků.</p>		
ZAK (CUS)	<p>SI.2.3 Prověření a získání souhlasu k specifikovaným požadavkům.</p> <p>Ověřit, že specifikace požadavků uspokojuje potřeby a dohodnout co zákazník očekává, včetně uživatelského rozhraní.</p>	<p><i>Specifikace požadavků [ověřená]</i> (Requirements Specification [verified])</p>	<p><i>Specifikace požadavků [odsouhlasená]</i> (Requirements Specification [validated])</p>

Analýza Softwarových požadavků

Cíle:	Cílem této činnosti je jasně definovat rozsah projektu a identifikovat klíčové požadavky na systém.
Odůvodnění:	Je důležité jasně definovat rozsah (hranice) projektu a identifikovat klíčovou funkcionalitu budoucího systému za účasti zákazníka, aby se předešlo budoucím problémům jako jsou zapomenuté klíčové funkce nebo požadavky.
Role:	Pracovní tým Zákazník
Artefakty:	Dokumentace požadavků
Postup:	<p>Step 1. Shromáždění informací o doméně aplikace (např. finance, zdravotnictví atd.)</p> <p>Step 2. Identifikace rozsahu projektu</p> <p>Step 3. Identifikace a získání požadavků</p> <p>Step 4. Struktura a prioritizace požadavků</p>
Popis postupu:	<p>Krok 1. Shromážděte informace o doméně</p> <p>V průběhu tohoto kroku pracovní tým zachycuje klíčové pojmy z oblasti podnikání zákazníka. Zákazník pomáhá pracovnímu týmu, tím že mu poskytuje veškeré informace (existující dokumentaci nebo vysvětlení) které mu napomáhají k porozumění této oblasti.</p> <p>Klíčové pojmy jsou uvedeny v části slovníku v osnově dokumentu Specifikace softwarových požadavků.</p> <p>Krok 2. Identifikace rozsahu projektu</p> <p>Softwarový analytik, kterému napomáhají osoby odpovědné za</p>

	<p>smluvní aspekty projektu (např. manažer prodeje), jasně identifikuj hlavní funkcionality, které zahrnuje rozsah projektu.</p> <p>Tip: Identifikace funkcí, které nejsou zahrnuty do projektu je také velmi důležitá, protože napomáhá objasnit rozdíly v chápání zákazníka.</p> <p>Krok 3. Identifikace a získání požadavků</p> <p>S ohledem na klíčové pojmy související se zákaznickou oblastí podnikání může analytik začít identifikaci požadavků. Žádná situace v IT projektech není totožná. V některých případech je většina požadavků již identifikovaná v dokumentu (např. ve výzvě k podání nabídek). Avšak ve většině případů jsou požadavky zákazníkům předávány ústně.</p> <p>Analytik musí identifikovat a sepsat seznam klíčových požadavků na systém k jeho vytvoření. V rámci tohoto kroku by analytik neměl začít detailně rozebírat tyto požadavky. Hlavním cílem je získat komplexní přehled systémových požadavků.</p> <p>Krok 4. Struktura a prioritizace požadavků:</p> <p>Za použití požadavků identifikovaných v předchozím kroku, analytik organizovat a strukturovat identifikované požadavky způsobem (např. podle byznys procesů, funkcí systému)</p> <p>Using requirements identified in the previous Step, the analyst has to organise and structure identified requirements accordingly (e.g. by business processes or by system functions).</p> <p>Dle priorit musí být zákazníkem identifikován systém klíčových funkcionalit. Priorita může být stanovena takto:</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 organizovány v Dokumentu Požadavků.</p>
--	--

4.1.3. Činnosti: SI.3 Identifikace softwarových komponent

Činnosti identifikace softwarových komponent transformují softwarové požadavky na architekturu systémových komponent. Tyto činnosti zahrnují:

- Pracovní tým přezkoumá projektový plán a určí rozdělení úkolů.
- Určí softwarové komponenty a související rozhraní.

Role	Seznam úkolů	Vstupní produkt	Výstupní produkt
PM PT (WT)	SI.3.1 Rozdělení úkolů členům pracovního týmu podle jejich rolí podle aktuálního projektového plánu.	<i>Projektový plan</i> (Project Plan)	
PT (WT)	SI.3.2 Porozumění specifikovaným požadavkům.	<i>Specifikace požadavků</i> (Requirements Specification)	
PT (WT)	SI.3.3 Vytvoření nebo aktualizace dokumentu Identifikace softwarových komponent. <i>Analyzujte specifikaci požadavků k vytvoření komponent, uspořádání do subsystémů a softwarových komponent, definujte vnitřní a vnější rozhraní systému.</i> Uvedte detaily softwarových komponent a jejich rozhraní pro jejich jednoznačné vytvoření.	<i>Specifikace požadavků</i> (Requirements Specification)	<i>Identifikace softwarových component</i> (Software Component Identification)

Identifikace softwarových komponent

Cíle:	Identifikace softwarových komponent, které budou odpovídat požadavkům zákazníka, bude je možné otestovat před jejich nasazením a bude možné ověřit, že jsou veškeré požadavky splněny.
Odůvodnění:	Identifikace softwarových komponent je klíčovým pilířem softwarového projektu. Chyby v popisu návrhu architektury, která bude zahrnovat všechny požadavky, je receptem na katastrofu. Zákazník nezaplatí, když návrh nebude odpovídat všem jeho požadavkům.
Role:	Pracovní tým
Produkt:	Specifikace požadavků Konfigurace softwaru
Artefakty:	UML or BPMN diagramy
Postup:	Krok 1. Porozumění specifikaci požadavků Krok 2. Vytvoření nebo aktualizace dokumentu Softwarových komponent
Popis postupu:	Krok 1. Porozumění specifikace požadavků <ul style="list-style-type: none"> ○ Prozkoumat každý požadavek a být si jistý, že jej každý chápe a je zařazen: <ul style="list-style-type: none"> ● Požadavky na funkce jsou logicky seskupeny.

	<ul style="list-style-type: none"> • Nefunkční požadavky jsou ve skupinách. ○ Ověřte, že je vše zařazeno a že rozumíte všem požadavkům. ○ V případě potřeby aktualizujte Specifikaci požadavků a přidejte nezbytné vysvětlení. <ul style="list-style-type: none"> • Uložte aktualizovaný dokument na úložiště. <p>Krok 2. Vytvoření nebo aktualizace dokumentu Softwarových komponent</p> <ul style="list-style-type: none"> ○ Analyzujte Specifikaci požadavků k vytvoření komponent, uspořádejte je do subsystémů a ke komponentám definujte vnitřní a vnější rozhraní. ○ Detailně popište vzhled a chování rozhraní, na základě specifikace požadavků se pokuste předvídat jaké budou potřeba zdroje pro implementaci. ○ Udělejte detail komponent a jejich rozhraní pro jejich jednoznačné vytvoření.
--	--

4.1.4. Činnost: SI.4 Tvorba softwaru

Činnost tvorba softwaru vytváří zdrojový kód softwaru a data softwarových komponent systému uvedených v SI.3. Činnosti zajišťují:

- Pracovní tým přezkoumává projektový plán a přerozděluje úkoly.
- Porozumění a identifikace softwarových komponent
- *Testovací scénáře, postupy pro jednotkové testování a integrační testování.*
- *Naprogramování softwarových komponent a aplikování jednotkových testů.*

Role	Seznam úloh	Vstupní produkt	Výstupní produkt
PM PT (WT)	SI.4.1 Rozdělení úkolů členům pracovního týmu podle jejich rolí podle aktuálního projektového plánu.	<i>Projektový plan</i> (Project Plan)	
PT (WT)	SI.4.2 Porozumění identifikovaným softwarovým komponentám	<i>Identifikace softwarových komponent</i> (Software Component Identification)	

Role	Seznam úloh	Vstupní produkt	Výstupní produkt
PT (WT)	SI.4.3 Tvorba nebo aktualizace softwarových komponent.	<i>Identifikace softwarových component</i> (Software Component Identification)	<i>Softwarové komponenty</i> (Software Components)
PT (WT)	SI.4.4 Vytvořit nebo aktualizovat testovací případy a postupy pro jednotkové a integrační testování založené na dokumentu Identifikace softwarových komponent. V případě potřeby, zákazník poskytne testovací data.	<i>Specifikace požadavků [ověřená]</i> (<i>Identifikace softwarových component</i> (Software Component Identification)	<i>Testovací scénáře a postupy</i> (Test Cases and Test Procedures)
PT (WT)	SI.4.5 Otestování softwarových komponent. Oprava chyb, dokud úspěšně neproběhnou veškeré jednotkové testy.	<i>Testovací scénáře a postupy</i> (Test Cases and Test Procedures) <i>Softwarové komponenty</i> (Software Components)	<i>Softwarové komponenty</i> (Software Components) <i>[Jednotkově otestované]</i> ([unit tested])

Tvorba softwaru

Cíle:	Vytvoření dříve identifikovaných softwarových komponent.
Odůvodnění:	Vývojáři mohou mít takovou sebedůvěru, že budou vytvářet komponenty bez systematického přístupu, i když by některé z nich mohly být použitelné pro tvorbu složitějších komponent. Existuje více přístupů k tvorbě komponent, zde využíváme Pseudokód, který je jedním z nejrozšířenějších. Postup byl přijat z Code Complete (viz odkaz).
Role:	Pracovní tým
Produkt:	Softwarové komponenty
Artefakty:	Softwarové komponenty
Postup:	Krok 1. Návrh komponent(y) Krok 2. Naprogramování komponent(y)

	Krok 3. Vyzkoušení komponent(y)
Popis postupu:	<p>Krok 1. Návrh komponent(y)</p> <p>a) Ověřte přínos komponenty. Pod tím se rozumí přínos komponenty k finálnímu produktu.</p> <p>b) Definujte problém, který má komponenta řešit. Podrobně popište, jaký problém bude komponenta řešit, aby mohla být následně vytvořena. V případě nedostatku informací k vytvoření komponenty, by jí vývojář měl dokončit společně s designerem.</p> <p>c) Požadovaná funkcionální může být dostupná ve standardních knihovnách. Zjistěte zda některé nebo všechny funkce této komponenty nejsou dostupné v knihovnách programovacího jazyka, v platformě, nebo nástrojích, které používáte.</p> <p>Pište pseudokód. Začněte od obecných věcí a postupně se dostávejte k více specifickým částem. Nejobecnější část komponenty je hlavička, která popisuje k čemu je komponenta určena. Nejdříve napište stručné prohlášení o účelu komponenty, dále prohlášení rozdělte do více nižších úrovní, tak abyste úplně pokryli oblast komponenty. Po vytvoření hlavního účelu komponenty popište správu uchyb, postup ošetření všech chyb komponenty, jako např. špatné vstupní hodnoty, špatné výstupní hodnoty vrácené z jiných částí atd.</p> <p>f) Návrh datové struktury komponenty. Definujte klíčové datové typy.</p> <p>g) Zkontrolujte pseudokód. Projděte pseudokód, když si něčím nejste jistí, pokuste se to vysvětlit např. Designerovi, toto vysvětlení vám pomůže utřídit si myšlenky a můžete vylepšit pseudokód.</p> <p>Krok 2. Naprogramování komponent(y)</p> <p>a) Napište deklaraci komponenty a vložte původní prohlášení z hlavičky do komentáře programovacího jazyka.</p> <p>b) Použijte původní pseudokód jako komentáře vysoké úrovně.</p> <p>c) Dopište zdrojový kód pod každým komentářem. Dopište zdrojový kód pod každou částí komentáře přeneseného z pseudokódu. Každý pseudokód popisuje blok nebo odstavec</p>

	<p>zdrojového kódu.</p> <p>d) Zkontrolujte, zda by kód neměl být dále rozdělen. V některých případech můžete mít mnoho kódu pod jednou částí pseudokódu. V tomto případě zvažte, zda nevytvořit dílčí komponentu.</p> <p>Step 3. Vyzkoušení komponenty</p> <p>V tomto kroku programátor přezkoumává kód, provádí kompilaci a debugging.</p> <p>a) Zkontrolujte kód</p> <ul style="list-style-type: none"> - Zkontrolujte program oproti požadavkům, zda byly všechny požadované funkce implementovány. - Projděte kontrolní seznam kódu, podle části 7, k nalezení veškerých chyb v programu. - Při kontrole označte všechny chyby ve zdrojovém kódu. - V návaznosti na kontrolu kódu opravte všechny chyby. - Po dokončení oprav vytvořte zdrojový program pro opravu chyb programu. - Zkontrolujte všechny opravy, abyste se ujistili, že byly správné. - Opravte všechny zbývající nedostatky. <p>b) Zkompilujte kód Nechte počítač zkontrolovat nedeklarované proměnné, konflikty v názvech, atd. Možná jste to udělali již před kontrolou kódu.</p> <p>c) Krokujte kódem prostřednictvím debuggeru Po rutinní kompilaci, použijte debugger a krokujte řádek po řádku. Ujistěte se, že se každý řádek provede tak, jak očekáváte. Toto můžete provádět po jednotkovém testu, abyste zjistili příčinu chyby.</p>
--	---

4.1.5 Činnost: SI.5 Softwarová integrace a testování

Činnosti Softwarové integrace a testování zajišťují, že integrované softwarové komponenty splňují definované požadavky na software. Činnosti zajišťují:

- Pracovní tým přezkoumá projektový plán a rozdělí si úkoly.
- Porozumění testovaným případům užití, testovacím procedurám a integračnímu prostředí.
- Integrované softwarové komponenty, opravené chyby a zdokumentované výsledky.

Role	Seznam úloh	Vstupní produkt	Výstupní produkt
PM PT (WT)	SI.5.1 Rozdělení úkolů členům pracovního týmu podle jejich rolí podle aktuálního projektového plánu.	<i>Projektový Plán</i> <i>(Project Plan)</i>	
PT (WT)	SI.5.2 Porozumění testovaným případům a procedurám. Vytvořte nebo aktualizujte testovací prostředí	<i>Testované případy užití a testovací procedury</i>	
PT (WT)	SI.5.3 Integrujte software za použití softwarových komponent, aktualizovaných testovaných případů a procedur pro integrační testování, pokud jsou potřeba.	<i>Softwarové Komponenty</i> <i>(Software Components)</i> <i>Testované případy užití a testovací procedury</i> <i>(Test Cases and Test Procedures)</i>	<i>Software</i> <i>Testované případy užití a testovací procedury (Test Cases and Test Procedures)</i> <i>[aktualizované]</i> <i>([updated])</i>
PT (WT)	SI.5.4 Proveďte softwarové testy za použití testovaných případů užití a testovacích procedur za účelem integrace a zdokumentujte výsledky testů v Testovacím Reportu.	<i>Software</i> <i>Testované případy užití a testovací procedury</i> <i>(Test Cases and Test Procedures)</i>	<i>Software [tested]</i> <i>Testovací Report</i> <i>(Test Report)</i>
PT (WT)	SI.5.5 Opravte nalezené chyby dokud se vše nepodaří úspěšně otestovat.	<i>Software</i> <i>Testovací Report</i> <i>(Test Report)</i> <i>Testované případy užití a testovací procedury</i> <i>(Test Cases and Test Procedures)</i>	<i>Software [opravený]</i> <i>([corrected])</i> <i>Test Report</i> <i>(Test Report)</i> <i>[Vady odstraněny]</i> <i>([defects eliminated])</i>
PT (WT)	SI.5.6 Použijte Specifikaci požadavků a software ke konfiguraci softwaru.	<i>Požadavky na softwarovou specifikaci</i> <i>(Requirements Specification Software)</i>	<i>Softwarová konfigurace</i> <i>(Software Configuration)</i> <i>- Specifikace Požadavků</i> <i>(Requirements Specification)</i> <i>- Software</i>

Softwarová integrace a testování

Softwarová integrace a testování	
Cíle:	K zajištění, že integrované softwarové komponenty splňují Požadavky na software.
Odůvodnění:	Tímto se umožní použití různých typů testů a díky tomu se identifikují problémy, které musí být odstraněny vývojovým týmem.
Role:	Zákazník
	Analytik
Artefakty:	Testovací report
Postup:	Krok 1: Připravte softwarové testovací prostředí
	Krok 2: Provedte testovací iterace
	Krok 3: Provedte regresní testy
	Krok 4: Ukončete testovací procedury
	Krok 5: Zdokumentujte výsledky v Testovacím Reportu
Popis postupu:	<p>Krok 1: Připravte softwarové testovací prostředí</p> <p>Během tohoto kroku, v testovacím prostředí; má být otestována dostupnost a připravenost komponent (opravené nebo baselined) a jsou připravena data, která jsou potřebná pro testování. Cílem je definovat řízené a nezávislé prostředí pro testování. Příprava testovacího prostředí zahrnuje např.: specifickou hardwarovou konfiguraci, operační systém, prohlížeč a TCP/IP konfiguraci, specifický systémový software, databázový engine a testing support engine.</p> <p>Krok 2: Provedte testovací iterace</p> <p>Před spuštěním je nezbytné zajistit, aby přidělený analytik rozuměl testům.</p> <p>Testovací iterace odpovídá prováděným testům. A testing iteration corresponds to the execution of tests. Testovací iterace se provede když vývojový tým provede jednotkové testování. Podle stability aplikace, může analytik rozhodnout, pro každou iteraci, zda spustí všechny připravené testy nebo jen jejich reprezentativní podmnožinu.</p> <p>Prostřednictvím prováděných zkoušek analytik zjistí odchylku od očekávaných hodnot, které jsou katalogizovány jako produktové vady. Produktové vady jsou roztříděny a zaznamenány.</p> <p>Pro klasifikaci chyb můžete vzít v úvahu:</p> <ul style="list-style-type: none"> • Blocker: Tento typ problému zastaví operaci programu/komponenty a produkuje výsledky které zabraňují chodu operace.

	<ul style="list-style-type: none"> • Testovací případ, který nemůže být proveden, protože předpoklady pro jeho spuštění nejsou splněny. • Funkční: To nastane v případě, kdy program provede operaci a výsledky neodpovídají očekávaným hodnotám. • Presentace: Otázky, které se týkají prezentace výrobku. Program musí být přizpůsoben definovaným gramatickým a ortografickým standardům jazyka, ve kterém je program předváděn. <p>Krok 3: Proveďte regresní testy</p> <p>Cílem je, ujistit se, že nový nebo upravený kód (řešící problémy během testů) je v souladu se stanovenými náležitostmi a nebyl ovlivněn při úpravách.</p> <p>Regresní testy zahájíte, jakmile technická skupina přijde s řešením zjištěných problémů nebo přidává nové vlastnosti do softwaru a ty jsou zavedeny do testovacího prostředí. Pro dosažení účinného postupu je možné kombinovat regresní testy a testovací iteraci.</p> <p>Krok 4: Ukončete testovací procedury</p> <p>Jakmile jsou definované iterace hotové a identifikované nedostatky byly vyřešeny, provede se analýza, že je udržitelná tendence k poklesu počtu nalezených chyb. Ukončení testování se provádí vytvořením závěrečné zprávy, postup testování, závěry a doporučení pro zlepšení procesu a výrobku. Tato doporučení tvoří základ pro poučení VSE. Vývojový tým rozhodne o uvolnění softwarového produktu do provozu na základě výsledku testů.</p> <p>Krok 5: Zdokumentujte výsledky v Testovacím Reportu</p> <p>Analytik nebo programátor musí připravit report o postupu testování, minimálně pro každou testovací iteraci. Zpráva musí obsahovat současný stav produktu. Zpráva musí být přezkoumána všemi členy týmu a musí připravit podklady pro nové testování.</p>
--	--

4.1.6. činnost: SI.6 Dodání produktu

Činnost dodání produktu poskytuje integrovaný softwarový produkt zákazníkovi. Činnost poskytuje:

- Zhodnocení Pracovního Týmu k *Projektovému Plánu* na určení přiřazení úkolů.
- Dodávku softwarového produktu a příslušné dokumentace v souladu s projektovým plánem.

Role	Seznam úkolů	Vstupy	Výstupy
PM PT _(WT)	SI.6.1 Přidělení úkolů členům Pracovních Týmů, které jsou v souvislosti s jejich rolí, podle aktuálního plánu projektu.	<i>Projektový Plán</i> (Project Plan)	
PT _(WT)	SI.6.2 Zhodnocení srozumitelnosti softwarové konfigurace.	<i>Software Konfigurace</i> (Software Configuration)	
PM PT _(WT)	SI.6.3 Provést doručení Projektovému Manažerovi a podpora doručení podle Projektového Plánu.	<i>Projektový Plán</i> (Project Plan) <i>Softwarová Konfigurace</i> (Software Configuration)	<i>Softwarová Konfigurace</i> [připraveno k dodání]

DODÁNÍ PRODUKTU

Cíle:	Zajištění pokračujících dodávek takových činností jako je ta na konci projektu. Každý výstup je k dispozici a splňuje kritéria pro přijetí definované v dodacích pokynů.
Odůvodnění:	Provedením pokračujících dodávek činností by nemělo vzniknout žádné překvapení, žádné zdržení k získání schválených výstupů. Jinak zákazník nedokončí platby vůči VME (Velmi malé entitě).
Role:	Pracovní Tým
Produkty:	Software Konfigurace Formulář akceptačního protokolu
Artefakty/výtvory:	Projektový Plán Formulář schválených dodacích pokynů Formulář akceptačního protokolu
Kroky:	Krok 1. Porozumění Softwarové Konfigurace. Krok 2. Provést dodávku podle Dodacích pokynů.
Popis kroků:	Krok 1. Porozumění Softwarové Konfigurace. <ul style="list-style-type: none"> • Získat z projektového plánu dodací podmínky • Vytvořit/získat potvrzenou specifikaci produktu <ul style="list-style-type: none"> ○ Analytik připraví výstupy (Analyst prepares the deliverables). ○ Analytik stanoví potvrzenou specifikaci pro konfiguraci produktu, včetně relevantního prostředí,

	<p>ručně navrženého dokumentu a konfigurovaného produktu.</p> <p>Krok 2. Provést dodávku podle Dodacích pokynů.</p> <ul style="list-style-type: none"> • Ověřte, že každá část softwaru splňuje kritéria pro přijetí. • Aktualizovat Formulář akceptačního protokolu • Naplánovat schůzku se zákazníkem • Získat schválení Formuláře akceptačního protokolu od zákazníka <ul style="list-style-type: none"> • Zákazník podepíše Formuláře akceptačního protokolu • Předat kopii Formuláře akceptačního protokolu zákazníkovi • Uložit Formulář akceptačního protokolu v repozitáři.
--	--

4.2 Popis rolí

Tohle je výpis rolí, jejich zkratk a popisu doporučených kompetencí.

	Role	Zkratka	Kompetence
1.	Zákazník (Customer)	ZAK (CUS)	Znalost procesů zákazníka a schopnost vysvětlit zákaznické požadavky. Zákazník (jeho představitel) musí mít pravomoc schvalovat požadavky a jejich změny. Pojem zákazník zahrnuje i uživatele, aby bylo zajištěno správné provozní prostředí Znalosti a zkušenosti v aplikační doméně.
2.	Projektový Manažer (Project Manager)	PM	Schopnost vedení se zkušenostmi při rozhodování, plánování, personální řízení, delegování a dohled, finance a vývoj software.
3.	Pracovní Tým (Work Team)	PT (WT)	Znalosti a zkušenosti na základě jejich rolí na projektu.

4.3. Popis produktu

Tohle je seznam vstupů, výstupů a interních procesů produktů, jejich popis, možné stavy a zdroje produktu. Zdroj může být jiný proces nebo externí entita pro projekt, jako je

například zákazník. Položky produktu v následujícím tabulkách jsou založeny na ISO/IEC15289 informační položky s několika výjimkami.

Status produktu dává informaci projektovému týmu o typu práce (úkolech) na produktu již hotových (například: vyhodnoceno, ověřeno, testováno, specifikačně potvrzeno). Tahle informace může být použita k započatí nových úkolů které mohou použít produkt jako vstup. Některé produkty nemají status přidělen protože jsou pouze informativní a nemění obsah (například: akceptační protokol, Registr oprav, záloha projektového repozitáře, ověřovací/validační výsledky).

	Jméno	Popis	Zdroj
1.	<i>Akceptační protokol</i> (Acceptance Record)	Dokumenty akceptace výstupů z projektu zákazníkem. Může mít následující charakteristiky: <ul style="list-style-type: none"> - Protokol o přijetí dodávky - Identifikuje datum příjmu - Identifikuje dodané prvky - Protokol ověřování výstupů zákazníkem podle zákaznické smlouvy. - Identifikuje jakékoliv otevřené problémy (je-li to možné) - Podpis zákazníka (odběratele) 	Projektové řízení (Project Management)
2.	<i>požadavek na změnu</i> (Change Request)	Identifikuje softwarový, či dokumentační problém, nebo žádané zlepšení a žádané modifikace. Může mít následující charakteristiky: <ul style="list-style-type: none"> - Identifikuje účel změny - Identifikuje požadovaný status (nový, přijato, odmítnuto) - Identifikuje požadované kontaktní údaje - Impacted system(s) - Dopad na provoz stávajících definovaných systémů - Dopad na související definované dokumentace - Kritičnost požadavku, datum potřeby Použitelné stavy jsou: přijato a sledováno.	Implementace Software (Software Implementation) Zákazník (Customer) Projektové řízení (Project Management)
3.	<i>Záznam z jednání</i> (Meeting Record)	Záznamy dohod vzniklých mezi Zákazníkem a/nebo Pracovním Týmem. Může mít následující charakteristiky: <ul style="list-style-type: none"> - Účel setkání - Účastníci - Datum, místo konání - Reference z předchozích setkání - Identifikuje vznesené otázky - Všechny otevřené otázky 	Projektové řízení (Project Management)

	Jméno	Popis	Zdroj
		<ul style="list-style-type: none"> - Dohody - Příští jednání, pokud bude. <p>Použitelný stav je: aktualizováno.</p>	
4.	<p><i>Záznam stavu postupu projektu</i> (Progress Status Record)</p>	<p>Zaznamenává status (stav) projektu proti plánu projektu. Může mít následující charakteristiky:</p> <ul style="list-style-type: none"> - Stav (status) aktuálních úkolů proti plánovaným úkolům - Stav aktuálních výsledků proti stanoveným cílům - Stav probíhajícího přidělování zdrojů proti plánovaným zdrojům - Stav skutečných nákladů proti rozpočtovým odhadům - Stav aktuálního času proti plánovanému harmonogramu - Stav aktuálního skutečného rizika proti dříve identifikovanému - Záznam jakýchkoliv odchylek od plánovaných úkolů a důvody proč k nim došlo. <p>Použitelný stav je: ohodnoceno.</p>	<p>Projektové řízení (Project Management)</p>
5.	<p><i>Projektový Plán</i> (Project Plan)</p>	<p>Prezentuje jak projektové procesy a činnosti budou provedeny, aby bylo projektu zajištěno úspěšné dokončení i kvalita dodávaných produktů. Zahrnuje následující prvky, které mohou mít následující charakteristiky :</p> <ul style="list-style-type: none"> - <i>Popis produktu</i> <ul style="list-style-type: none"> o Účel o Obecné zákaznické požadavky - Popis rozsahu toho, co je v ceně a co ne - Dodávky - seznam produktů jenž má být dodán zákazníkovi - Úkoly včetně ověřování, validace a zhodnocení se Zákazníkem a Pracovním Týmem, tak aby byla zajištěna kvalita pracovních produktů. Úkoly mohou být zastoupeny jako struktura <p>Diagram rozkladu prací .</p> <ul style="list-style-type: none"> - Vztah a Závislost úkolů - Odhadovaná doba trvání úkolů - Zdroje (lidé materiály, standardy, vybavení a nástroje) a plán, kde je potřeba zdrojů. 	<p>Projektové řízení (Project Management)</p>

	Jméno	Popis	Zdroj
		<ul style="list-style-type: none"> - Složení Pracovního Týmu - Plán projektových úkolů, očekávané zahájení a datum dokončení pro každý úkol. - Odhad úsilí a nákladů - Identifikace rizik projektu <p>Použitelné stavy jsou: přijmuto</p>	
6.	<i>Projektový repozitář</i> (Project Repository)	<p>Elektronický zásobník na skladování projektové práce produktů a hotových částí k dodání. Může mít následující charakteristiky:</p> <ul style="list-style-type: none"> - Uskladňovat pracovní produkty projektu - Uskladňovat vydané připravené produkty - Skladovací a obnovovací schopnosti - Schopnost procházet obsah - Výpis obsahu s popisem atributů - Sdílení a předávání pracovních produktů mezi pracovními týmy. - Efektivní kontrola nad přístupem - Udržujte popisy pracovních produktů - Obnova archivních verzí pracovních produktů - Schopnost nahlásit stav (status) pracovního produktu - Změny pracovních produktů jsou sledovány požadavkem na změnu <p>Použitelné stavy jsou: aktualizováno.</p>	Projektové řízení (Project Management)
7.	<i>Specifikace Požadavků</i> (Requirements Specification)	<p>Identifikuje požadavky na software. Může mít následující charakteristiky:</p> <ul style="list-style-type: none"> - Úvod-obecný popis softwaru a jeho využívání v rámci zákaznického biznisu; - Popis požadavků: <ul style="list-style-type: none"> - Funkcionalita - stanovená potřebuje být splněna pomocí softwaru, při použití v konkrétních podmínkách. Funkcionalita musí být přiměřená přesná a bezpečná. - Uživatelské rozhraní - definice těch vlastností uživatelského rozhraní, které umožňují porozumět a naučit se software snadno, tak že uživatel bude moci vykonávat jeho / její úkoly efektivně včetně ukázkového popisu rozhraní ; - Vnější (Externí) rozhraní - definice 	Implementace Software (Software Implementation)

	Jméno	Popis	Zdroj
		rozhraní s jiným softwarem nebo hardwarem; Každý požadavek je identifikovaný, jedinečný a je ověřitelný nebo může být posouzen. Použitelné stavy jsou: ověřeno a potvrzeno.	
8.	<i>Software</i>	Softwarová položka (zdrojový a spustitelný kód softwaru) pro zákazníka, tvořen kolekcí integrovaných softwarových komponent. Použitelné stavy jsou: testováno	Implementace Software (Software Implementation)
9.	<i>Softwarová Komponenta</i> (Software Component)	Sada souvisejících jednotek kódu. Použitelné stavy jsou: jednotka otestovaná	Implementace Software (Software Implementation)
10.	<i>Softwarová Konfigurace</i> (Software Configuration)	Jednoznačně identifikovaný a konzistentní sada softwarových produktů zahrnuje : - <i>Specifikaci požadavků</i> - <i>Software</i> Použitelné stavy jsou: dodáno a potvrzeno.	Implementace Software (Software Implementation)
11.	<i>Identifikace Softwarové Komponenty</i> (Software Component Identification)	Textová a grafická informace o softwarové struktuře. Tato struktura může obsahovat následující části: Popisuje celkovou Softwarovou strukturu: - Identifikuje požadované softwarové komponenty - Identifikuje vztah softwarových komponent mezi sebou	Implementace Software (Software Implementation)
12.	<i>Specifikace prací</i> (Statement of Work)	Popis prací kterou je třeba udělat a souvisí s vývojem software. Může zahrnovat: - Popis produktu o Účel o Obecné Požadavky zákazníků - Popis rozsahu toho, co je v ceně a co ne - Výstupní seznam produktů jenž mají být dodány zákazníkovi Použitelný stav je: přezkoumáno.	Zákazník (Customer)
13.	<i>Testovací případy a Testovací Procedury</i> (Test Cases and	Prvky potřebné k testování kódu. Testovací případ může zahrnovat: - Identifikuje testovací případ - Zkušební položky - Vstupní údaje	Implementace Software (Software Implementation)

	Jméno	Popis	Zdroj
	<i>Test Procedures)</i>	<ul style="list-style-type: none"> - Výstupní parametry - Environmentální potřeby - Zvláštní procesní požadavky - Závislosti rozhraní <p>Zkušební postupy mohou zahrnovat:</p> <ul style="list-style-type: none"> - Identifikuje: jméno testu, popis testu a datum dokončení testu - Identifikuje potenciální problémy s implementací - Identifikuje osobu, která dokončila testovací proceduru - Identifikuje předpoklady - Identifikuje kroky procedury, včetně čísla kroku , požadovanou akci dle testera a očekávané výsledky 	
14.	<i>Testovací Report</i> (<i>Test Report</i>)	<p>Dokumentuje provádění testů . Může zahrnovat:</p> <ul style="list-style-type: none"> - Shrnutí o každé vadě - Identifikuje testera, který našel danou vadu - Identifikuje ovlivněnou funkci (e) pro každou vadu - Označuje datum, kdy každá vada vznikla - Označuje datum, kdy byla vyřešena daná vada - Identifikuje osobu, která vyřešila danou vadu 	Implementace Software (Software Implementation)

4.4. Popis artefaktů

Artifacts	Definition
Akceptační dokument (Acceptance document)	Dokument, který stanovuje zákaznickou akceptaci dodaných výstupů vycházejících z projektu.
Projektový popis (Project Description)	Vysoká úroveň popisu projektu zahrnující; rozsah; cíle a hlavní výstupy.
Projektový Plán (Project Plan)	Prohlášení o tom, jak a kdy má být dosaženo projektových Cílů s ukazáním na hlavní produkty, milníky, činnosti a zdroje potřebné pro projekt.
Dokument Požadavků (Requirements Document)	Dokument, v němž jsou centralizované všechny identifikované požadavky.
Software	Ucelený soubor softwarových produktů, které obsahují: <ul style="list-style-type: none"> • Specifikaci požadavků • Softwarové komponenty • Software (jednotku, produkt, položku) • Zkušební a zpráv o mimořádných událostech • Pracovní manuál • Návod k obsluze
<i>Softwarová Komponenta</i> (<i>Software Component</i>)	Sada souvisejících jednotek kódu. Použitelné stavy jsou: jednotka otestovaná, opravena a specifičtěně potvrzena.
<i>Softwarová Konfigurace</i> (<i>Software Configuration</i>)	Jednoznačně identifikovaná a konzistentní sada softwarových produktů, jenž zahrnuje : <ul style="list-style-type: none"> - <i>Specifikaci Požadavků</i> - <i>Software Design</i> - <i>Softwarové Komponenty</i> - <i>Software</i> - <i>Testovací Reporty</i> - <i>Provozní Příručka Produktu</i> - <i>Uživatelská Dokumentace Softwaru</i> - <i>Provozní Dokumentace</i>
<i>Testovací Report</i> (<i>Test Report</i>)	Záznam výsledků provedených testování ve specifickém/konkrétním prostředí
UML Diagramy (<i>UML Diagrams</i>)	Tyto diagramy usnadní grafické znázornění návrhu pomocí UML (Unified Modeling Language). UML může poskytnout jednoduchý a standardní způsob, jak vyjádřit data, procesy nebo architekturu.

5. Šablony a nástroje

Šablony jsou poskytnuty jako příklady, které mohou být upraveny pro Váš projekt.

5.1. Softwarová implementace - vstupní šablona

Obsah šablony – Oznámení o setkání

1. Název setkání (zahrnující jméno projektu)
2. Důležitost setkání
3. Adresy
4. Cíl setkání
5. Obsah sdělení
6. Datum, místo a čas
7. Další (např. jídlo v ceně)

Pomocný seznam šablony

číslo	Člen týmu Jméno	Role	Asistence		Podpis
			Ano	Ne	
Komentáře					

5.2. Šablona - Analýz softwarových požadavků

Tabulka požadavků

Může být použita struktura v Excelu, např.

ID	Požadavek	Popis	Priorita		
			Vysoká	Střední	Nízká

Každý požadavek lze definovat jako tabulku. Struktura tabulky je následující:

Projekt	<Identifikace projektu>
Datum	<Datum poslední změny>
Požadavek	<Identifikace, text požadavku>
Popis	<Detailní popis požadavku >

Odůvodnění	<Proč je požadavek důležitý>
Priorita	<Priorita vyřízení požadavku>
Závislost	<Vztahy s dalšími požadavky >
Komentář	<další poznámky>

5.3 Šablona – Identifikace softwarových komponent

Odkazy na standardy diagramů.

Šablona	Zdroj
UML	http://www.uml.org
BPMN	http://www.bpmn.org

Může být použita struktura excelu, např.:

Obsah šablony – Identifikace komponent

Identifikace projektu (Jméno, Datum)

1. Komponenta 1.
 - 1.1. Identifikace
 - 1.2. Popis
 - 1.3. Sub-komponenty
 - 1.4. Požadavky
2. Component 2.
 - 2.1. Identifikace
 - 2.2. Popis
 - 2.3. Sub-komponenty
 - 2.4. Požadavky
3. ...

5.4 Tvorba softwaru

Odkazy na šablony standardů kódů.

Šablona	Zdroj
General coding Standard template	http://www.construx.com
Zdrojový kód template C / C++	http://www.construx.com
Zdrojový kód template Java	http://www.construx.com
Coding Standards Java	http://www.ontko.com/java/java_coding_standards.html
Zdrojový kód template Visual Basic	http://www.construx.com
Zdrojový kód template XML	http://www.construx.com

Odkazy na šablonu specifikace uživatelského prostředí

Šablona	Zdroj
Specifikace uživatelského prostředí	http://www.hcirn.com/tutor/docs/uitempl.php

5.5 Šablona - Specifikace testů

Může být použita struktura excelu, např.:

Typ aplikace	
Provozní režim	
Témata	
Cílová skupina	
Typy médií, které budou použity	

Klientův počítač	Charakteristika softwaru	Verze
Operační systém		
Prohlížeč		
Požadavky na prohlížeč		
Komunikační protokol		
	Charakteristika hardwaru	Kapacita
Procesor		
Hard disk		
RAM		
Rozlišení obrazovky		
Multimedia		
Rychlost CD-ROM		
Videokarta		
Zvuková karta		
3D akcelerátor		
	Další hardwarové charakteristiky klienta	Kapacita
Tiskárna		
Barcode Scanner		

Klientský počítač	Charakteristika softwaru Web serveru	Verze
Operační systém		
Web server		
Programovací jazyk		
Jazykové požadavky		
	Charakteristika softwaru databázového serveru	Verze
Operační systém		
Databáze		
	Charakteristika hardwaru webového serveru	Kapacita
Procesor		
Hard Disk		
RAM		

Obsah šablony – Test Report

1. Cíl testu
2. Výsledky testu
a. Požadavek na ověření či selhání
3. Komentáře a doporučení
a. Kritické chyby
b. Závažné poruchy
c. Malé chyby
4. Integrace komponent	
a. Test log	
b. Dokumentace testu	
c. Tabulky testovaných dat	
d. Analýza a závěry
5. Raw testovací datový soubor
6. Obecné závěry

5.6 Šablona – Dodávka produktu

Formulář instrukce dodání

Formulář instrukce dodání									
Identifikace projektu nebo Název zákazníka: _____									
Vytvořil (Name Initial): _____									
Datum (yyyy-mm-dd): _____									
Identifikace dodávek (např. hardware, software, documentace atd.):									
Dodací podmínky:									
Sekvenční řazení úkolů, jak mají být provedeny:									
Použitelné verze:									
Akceptační kritéria:									
Akceptační kritéria	Datum splnění criteria (yyyy-mm-dd)								
1									
2									
3									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Softwarové komponenty</th> <th style="width: 50%;">Informace o verzi</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Softwarové komponenty	Informace o verzi							
Softwarové komponenty	Informace o verzi								
Záloha a obnovení:									
Schváleno:									

_____ Projektový manažer	_____ Zákazník nebo zástupce zákazníka
Datum (yyyy-mm-dd): _____	

Formulář - akceptační protokol

Formulář – akceptační protokol			
Identifikace projektu:			
Vytvořil (Name Initial):			
Datum (yyyy-mm-dd):			
Akceptační kritéria:			
Akceptační kritéria	Datum splnění kritéria (yyyy-mm-dd)		
1			
2			
3			
4			
5			
Podepsání převjímky:			
Softwarové komponenty	Informace o verzi	Datum převzetí zákazníkem	Podpis zákazníka

Podpisy:

Projektový manažer

Zákazník nebo zástupce zákazníka

Datum (yyyy-mm-dd): _____

Nástroje

Tato sekce informuje o nástrojích, které mohou pomoci podpořit některé aktivit procesů softwarové implementace.

Softwarová integrace a testy

Pro zajištění kontroly nad otázkami k produktu je nutné zavést systém, nejlépe web systém, který umožňuje průběžné hlášení problémů produktu a vyřizování žádostí, které jsou s nimi spojené.

Jako klíčové prvky, musí tento systém umožňovat:

- Automatické přiřazení otázek
- Připojení dokumentů
- Generování automatických upozornění na email.
- Řízení změn
- Generování jednoduchých interaktivních dotazů
- Problém klasifikace: Typ, případ, závažnost

Existuje několik open sourceových testovacích nástrojů dostupných na internetu, zde je web, který některé z nich nabízí:

<http://www.opensourcetesting.org/functional.php>

Dodání produktu

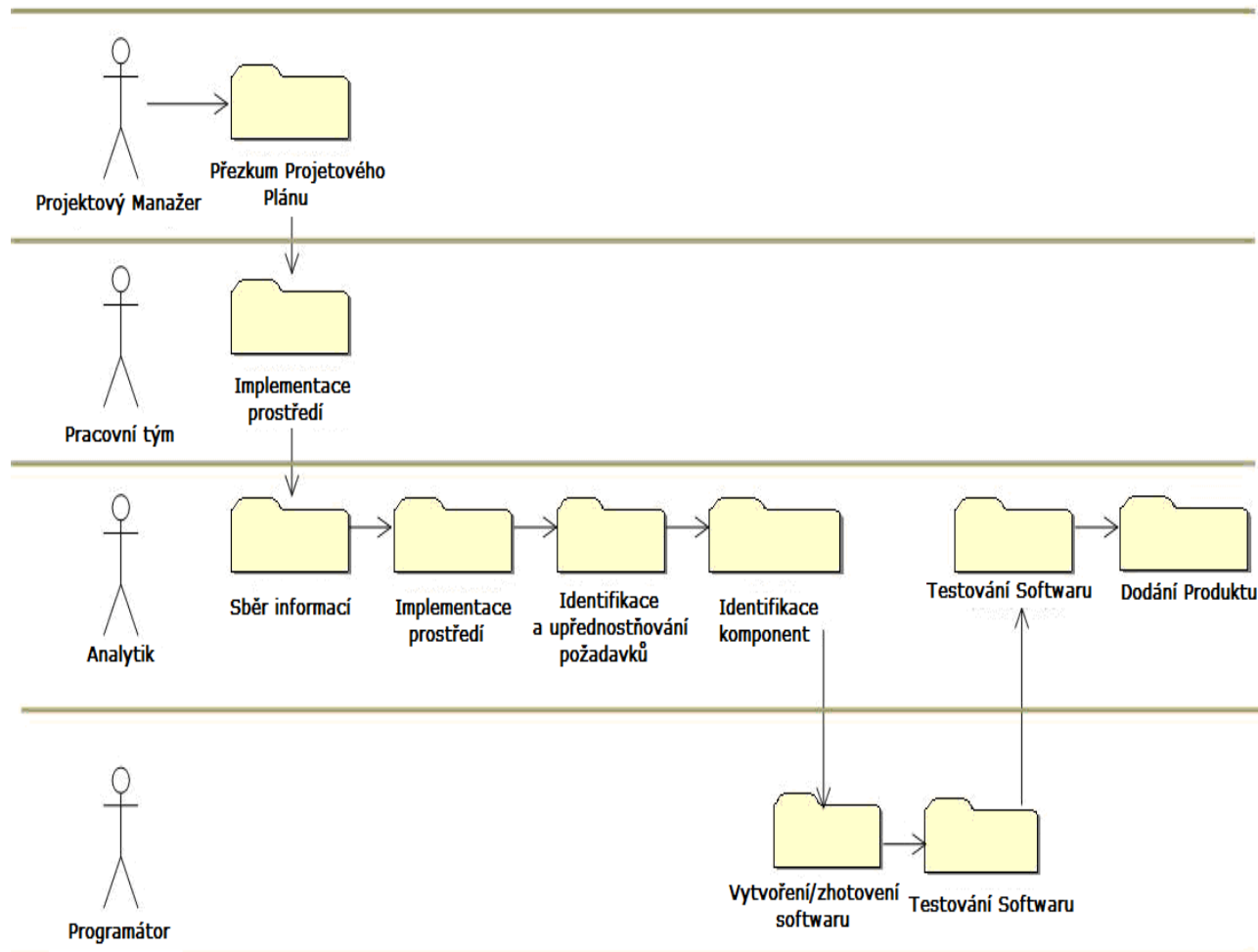
- Nástroje pro správu verzí jako CVS (Concurrent Versions System) nebo SVN (version management).
- Jsou dostupná open source úložiště např. Google Doc
- Nástroje pro Configuration Management

6. Příklad činnosti životního cyklu

Prohlášení: Tato část obsahuje některé grafické znázornění příkladů životního cyklu testování software postupů. Tyto příklady jsou poskytovány aby pomohly čtenáři při provádění jeho vlastního cyklu testování softwaru vyhovujícím kontextu a omezení jeho IT projektu.

Příklad životního cyklu softwarové implementace

Toto je příklad - použijte SPEM šablonu pro Microsoft Visio (<http://www.pa.icar.cnr.it/cossentino/FIPAmeth/docs/SPEM.vss>) za účelem vytvoření takového schématu.



Obrázek 3 Příklad životního cyklu softwarové implementace

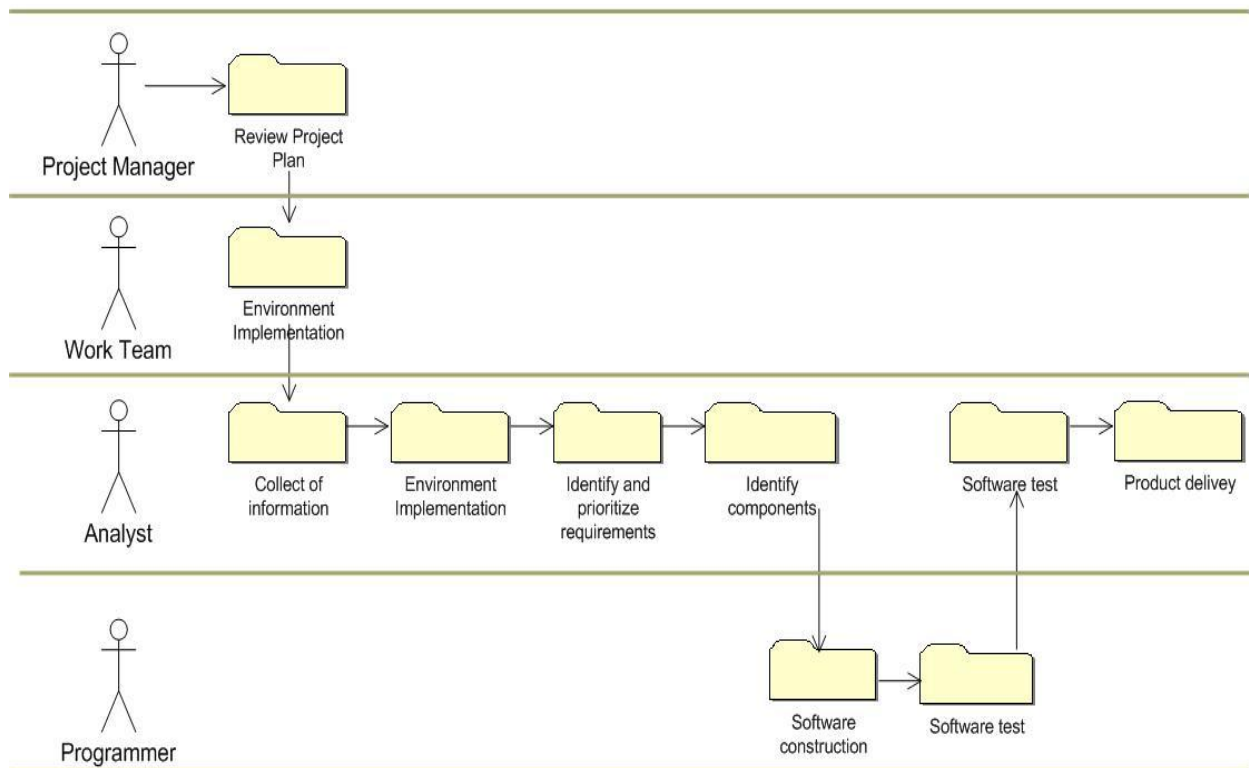


Figure 3 Example of Software Implementation Lifecycle

7. Kontrolní seznam

7.1 Kontrolní seznam Analýzy softwarových požadavků

Tento kontrolní seznam požadavků je založen na [Constr07]

RS 1 Testovatelné	Všechny požadavky jsou ověřitelné (objektivně)
RS 2 Kompletní	Jsou požadavky úplné?
RS 3 Opravit	Požadavky musí být v pořádku (odrážet přesně požadavky zákazníka)
RS 4 Unikátní	Požadavky musí být uvedeny pouze jednou
RS 5 Základní	Požadavky musí být rozděleny do jejich nejzákladnější formy
RS 6 Rozsah	Jsou požadavky v rozsahu působnosti?
RS 7 High Level	Požadavek musí být uvedený v podmínkách konečné potřeby, nesmí být chápán z významu (řešení)
RS 8 Jednoznačný	Požadavky musí být uvedeny v podmínkách, které lze interpretovat pouze jedním způsobem.

7.2 Kontrolní seznam Identifikace softwarové komponenty

RS 1 Testovatelné	Všechny komponenty jsou ověřitelné (objektivně)
RS 2 Kompletní	Jsou komponenty úplné?
RS 3 Opravit	Komponenty musí být v pořádku (odrážet přesně požadavky zákazníka)
RS 4 Unikátní	Komponenty musí být uvedeny pouze jednou
RS 5 Základní	Komponenty musí být rozděleny do jejich nejzákladnější formy
RS 6 Spojit	komponenty a Sub-Komponenty jsou propojeny
RS 7 Přidružit	Všechny požadavky musí být přidružené s komponenty.
RS 8 Jednoznačný	Komponenty musí být uvedeny v podmínkách, které lze interpretovat pouze jedním způsobem.

7.3 Kontrolní seznam konstrukce software

Kontrolní seznam přezkumu kódu

Poznámka: Tento Kontrolní Seznam může být přizpůsoben k aplikování na jazyk, ve kterém programujete

Subject	Description
Kompletní (Complete)	- Ověřte, že všechny funkce v designu jsou kódovány a že všechny potřebné funkce a postupy byly implementovány.
Logika (Logic)	- Ověřte, že tok programu a všechny procedury a funkční logika je v souladu s detailním designem.
Smyčky (Loops)	- Udělejte ruční simulaci všech smyček a rekurzivních postupů, které nebyly simulovány v detailním přezkoumání návrhu nebo inspekci. - Ujistěte se, že každá smyčka je správně zahájena i ukončena. - Zkontrolujte, že každá smyčka je spuštěna správný počet opakování.
Volání (Calls)	- Zkontrolujte všechny funkce a volání procedur, aby jste se ujistili, že přesně odpovídají definici formátů a typů.
Deklarace (Declarations)	Ověřte každou proměnnou a parametr že: - má přesně jednu deklaraci - se používá pouze v její deklarované působnosti - je napsána správně tam, kde se používá
Inicializace (Initialization)	- Zkontrolujte, že každá proměnná je inicializována.
Limity (Limits)	- Zkontrolujte, zda všechny proměnné, pole a indexy, abychom zajistily, že jejich použití není delší než deklarované limity.
Begin-end	- Zkontrolujte všechny begin-end dvojice, jejich ekvivalenty, včetně případů vnoření proti překroucení/chybné interpretaci.
Boolean	- Zkontrolujte Boolean podmínky
Formát (Format)	- Zkontrolujte každý řádek programu, formát instrukce, syntaxi a interpunkce.
Ukazatele (Pointers)	- Zkontrolujte, zda jsou všechny ukazatele správně použity.
Vstupy-výstupy (Input-output)	- Zkontrolujte všechny vstupní a výstupní formáty.
Pravopis (Spelling)	- Zkontrolujte, že každá proměnná, parametr a klíčová práce je správně napsána.
Komentáře (Comments)	- Ujistěte se, že všechny komentáře jsou přesné a podle normy.

7.4 Kontrolní seznam Softwarové Integrace a Testů

Informace by měly být převzaty z Construx¹
<http://www.construx.com/Page.aspx?nid=208>

7.5 Kontrolní seznam Dodání produktu

Typická akceptační kritéria pro VSE

- Produkt podporující média je správně označen, při nejmenším ukazuje název produktu, datum vydání, a správné číslo verze.
- Označení výrobku, zahrnuje dodací lokaci/umístění a dosažení personálu akceptace produktu (je-li to možné).
- Software je generovaný z projektového úložiště v souladu s dodacími pokyny
- Software který má být dodán je nejnovější verzí softwaru z projektového úložiště
- Dokument Popisu Verze byl zkontrolován.
- Dokument Popisu Verze je součástí podpůrných médií (supporting media).
- Všechny testy byly provedeny úspěšně
- Všechny chyby byly opraveny
- Všechny dokumentace byly aktualizovány
- Uživatelská příručka byla zkontrolována.
- Uživatelská příručka je dodávána s podpůrnými médii.
- Všechna témata byly pokryty ve schválených Formulářích Dodacích Instrukcí/Pokynů
- Formulář Akceptační Protokolu byl aktualizován a je připraven k podpisu
- Informace potřebné k dodání (například: adresa, zástupce zákazníka) byly ověřeny před dodáním
- Váš zákazník byl informován kdy bude dodávka uskutečněna
- Zákazník Vás informoval, že všechny jeho přípravy na dodávku jsou hotovy

Typická akceptační kritéria pro Zákazníka

- Dodané softwarové komponenty splňují schválené dodací instrukce
- Dodané softwarové komponenty splňují požadavky

¹ <http://www.construx.com>

Verze 0.3

- Očekávané dokumentace, testovací záznamy, designové informace, školící materiály atd. byly dodány
- Dodané softwarové komponenty pracují v určitém prostředí (jak technickém, tak organizačním), které je pro uživatele přijatelné
- Formulář Akceptačního Reportu je přezkoumán a podepsán

8. Reference

Klíč	Reference
[MCC 04]	Steve McConnell, <i>Code Complete</i> , Second Edition, Redmond, Wa.: Microsoft Press, 2004.
[IEEE 1012-2004]	IEEE 1012-2004 IEEE Standard for Software <i>Verification and Validation</i>
[ISO/IEC 12207]	ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes.
[ISO/IEC 15289]	ISO/IEC 15289:2011 Systems and software engineering - Content of systems and software life cycle process information products (Documentation)
[ISO/IEC 29110]	ISO/IEC TR 29110-5-1-1:2012, Software Engineering—Lifecycle Profiles for Very Small Entities (VSEs) – Part 5-1-1: <i>Management and Engineering Guide</i> —Generic profile group: <i>Entry Profile</i> . ISO/IEC TR 29110-5-1-1:2012 is available at no cost on the following ISO site: http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html
[ISO/IEC/IEEE 24765]	ISO/IEC/IEEE 24765:2010 Systems and software engineering vocabulary An electronic version of the glossary is available at: http://pascal.computer.org/sev_display/index.action
[IEEE 1233-1998]	IEEE Guide for Developing System Requirements Specifications
[PMI 2008]	A Guide to the <i>Project Management</i> Body of Knowledge (PMBOK), <i>Project Management</i> Institute, 2008.
[crm.com]	http://searchcrm.techtarget.com/ . Consulted July 10, 2010.

9. Hodnotící formulář

<p>Implementační balíček – Implementace Softwaru – Vstupní profil V 0.3</p> <p>Vaše zpětná vazba nám umožní zlepšit tento implementační balíček; vaše komentáře a doporučení jsou vítány.</p>	
<p>1. Jak jste spokojen s obsahem tohoto implementačního balíčku?</p> <p><input type="checkbox"/> <i>Velmi spokojený</i> <input type="checkbox"/> <i>Spokojený</i> <input type="checkbox"/> <i>Ani spokojený, ani nespokojený</i> <input type="checkbox"/> <i>Nespokojený</i> <input type="checkbox"/> <i>Velmi nespokojený</i></p>	
<p>2. The sequence in which the topics are discussed, are logical and easy to follow?</p> <p><input type="checkbox"/> <i>Velmi spokojený</i> <input type="checkbox"/> <i>Spokojený</i> <input type="checkbox"/> <i>Ani spokojený, ani nespokojený</i> <input type="checkbox"/> <i>Nespokojený</i> <input type="checkbox"/> <i>Velmi nespokojený</i></p>	
<p>3. Jak jste byl spokojen se vzhledem a formátem tohoto implementačního balíčku?</p> <p><input type="checkbox"/> <i>Velmi spokojený</i> <input type="checkbox"/> <i>Spokojený</i> <input type="checkbox"/> <i>Ani spokojený, ani nespokojený</i> <input type="checkbox"/> <i>Nespokojený</i> <input type="checkbox"/> <i>Velmi nespokojený</i></p>	
<p>4. Byla zahrnuta některá zbytečná témata? (prosím popište)</p>	
<p>5. Jaké chybějící téma byste rád viděl v tomto balíčku? (prosím popište)</p> <ul style="list-style-type: none"> • Navrhované téma: • Odůvodnění nového tématu: 	
<p>6. Nějaké chyby v implementačním balíčku?</p> <ul style="list-style-type: none"> • Uvedte prosím: <ul style="list-style-type: none"> • Popište chyby : • Místo chyby (sekce #, číslo #, tabulka #) : 	
<p>7. Other feedback or comments:</p>	
<p>8. Doporučil byste tento implementační balíček kolegovi z jiného Velmi malé entity?</p> <p><input type="checkbox"/> <i>Určitě</i> <input type="checkbox"/> <i>Možná</i> <input type="checkbox"/> <i>Nejsem si jistý</i> <input type="checkbox"/> <i>Probably Not</i> <input type="checkbox"/> <i>Definitely Not</i></p>	

Optional

- Jméno: _____
- e-mail: _____

Odešlete tento formulář na: claude.y.laporte@etsmtl.ca