

Modely softwarových procesů

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	LS 2014/15
Autoři – jméno, příjmení, xname	Martin Baláž – xbal34 Adam Brousek – xbroa03
Téma	Modely softwarových procesů
Datum odevzdání	14. 5. 2015

Abstrakt

Práce popisuje některé typy modelů softwarových procesů, konkrétně referenční modely a vybrané modely životního cyklu. U referenčních modelů vychází z normy ISO/IEC TR 24774 a v ní definované náležitosti ukazuje na konkrétním příkladu referenčního modelu. Dále je v práci ukázáno, jak je možné za základě účelu a výstupů procesu namodelovat softwarový proces využitím přístupu composition tree a postup je ukázán na Configuration Management procesu definovaném v normě ISO/IEC 12207:2000. A na závěr jsou uvedeny tři základní a historicky nejdůležitější modely procesů životního cyklu softwaru, a to Vodopádový model, V-model a Spirálový model.

Klíčová slova

model, software, proces, referenční, vodopádový, spirálový, V-model, životní cyklus, composition tree, Norma ISO/IEC TR 24774

Obsah

1	Úvod.....	4
2	Norma ISO/IEC TR 24774 [4].....	5
2.1	Title (název).....	5
2.2	Purpose (účel).....	5
2.3	Outcomes (výstupy).....	6
2.4	Activities (aktivita).....	6
2.5	Tasks (úkoly).....	7
2.6	Information items (informační položky).....	7
3	Composition tree.....	8
3.1	Vysvětlení Composition tree.....	8
3.2	Využití přístupu CT k modelování softwarových procesů.....	9
4	Modely procesů životního cyklu softwaru.....	12
4.1	Vodopádový model (Waterfall).....	12
4.2	V-model.....	13
4.3	Spirálový model.....	14
5	Závěr.....	17
6	Zdroje.....	18

1 ÚVOD

Modely softwarových procesů jsou důležité pro kvalitní tvorbu procesů v organizaci. Používání modelů pomáhá ušetřit čas a tvořit procesy na základě již ověřených metod. Tato práce se blíže bude zabývat otázkou správného definování modelu procesu a jeho elementů a popíše jak lze následně proces namodelovat pomocí composition tree.

Nakonec také v krátkosti shrneme modely životního cyklu, které do modelů softwarových procesů také spadají, ale tento text se primárně zaměřuje na výše popsané, tudíž jejich zpracování bude spíše informativního charakteru.

Cílem práce bude vysvětlit jak správně definovat model procesu a jeho elementy dle platné normy ISO/IEC TR 24774 a jak lze využít composition tree při modelování softwarových procesů. Oboje včetně názorného příkladu.

2 NORMA ISO/IEC TR 24774 [4]

Tato norma vydaná v roce 2010 definuje vzory pro procesní popis v referenčních modelech. Jejím dodržování umožňuje společnosti efektivnější sdílení, propojování a kontrolování procesů díky unifikaci procesních popisů. Obsahem je výčet procesních elementů a popis způsobu jejich užití. V následujících podkapitolách je uveden výčet těchto elementů včetně jejich zjednodušeného popisu.

2.1 TITLE (NÁZEV)

Název procesu je podstatné jméno, nebo sousloví, které vystihuje proces. Je důležité používat podstatná jména, a ne slovesa, která se snaží shrnout průběh procesu a mohou tak více svádět k špatné interpretaci. Název by měl proces jasně odlišovat od ostatních procesů v modelu, proto je někdy třeba v průběhu modelování změnit název procesu, aby více konkretizoval dva podobné procesy.

Název může popisovat jak obecný proces (proces vykreslení), tak konkrétní instanci obecného procesu v určitých podmínkách (proces vykreslení v projektu X). Instance by měla dětit název procesu ze kterého je odvozena a tato zděděná část může být odlišena v názvu například kurzívou.

2.2 PURPOSE (ÚČEL)

Účel popisuje konečný cíl průběhu procesu. V případech, kde by se mohly procesy překrývat by účel měl definovat hranice popisovaného procesu.

Účel by měl být zachycen v ideálním případě jednou větou, která začíná slovy: "Cílem procesu je..". Použití spojky "a" je třeba se v této větě vyhýbat, jinak by popis účelu byl agregací několika výstupů a ne jasným určením cíle.

2.3 OUTCOMES (VÝSTUPY)

Výstupy popisují pozorovatelné výsledky úspěšně dokončeného procesu. Jsou měřitelné a konkrétní, technické nebo obchodní a jsou pozorovatelné a hodnotitelné. Měly by být odlišeny od výhod, které jsou pozitivní playnouchými z průběhu procesu. Výhody obvykle nejdou vyhodnotit a ač mohou poskytovat jistou motivaci k provádění procesu nejsou hlavním důvodem k jeho provedení.

- výčet výstupů musí začínat slovy: “V důsledku úspěšného provedení tohoto procesu..”
- výčet by se měl skládat z oznamovacích vět v přítomném čase
- výstupy by měly být vyjádřeny jako pozitivní a pozorovatelné mety (např.: dosažení stavu, udržení stavu, výroba produktu)
- výstupy by neměli mít větší rozsah než dva řádky a přibližně 20 slov
- jeden proces by měl mít od 3 do 7 výstupů
- výstup by měl mít jeden jediný výsledek, proto je třeba předcházet používání spojky “a”
- jako test kompletnosti by soubor výstupů měl být dostačující pro dosažení stanoveného účelu procesu
- jako test relevance by měl být každý výstup formulován tak, že je k dosažení účelu p

2.4 ACTIVITIES (AKTIVITY)

Aktivity jsou výčet akcí potřebných k dosažení výstupů. Jsou to konstrukce pro seskupování souvisejících úkolů.

Je třeba se vyvarovat definování časových úseků a dob trvání, jelikož ty se mohou měnit podle životního cyklu. Pokud je to ovšem nutné, je důležité to explicitně uvést. Aktivity by také neměly být považovány za kroky v určitém postupu, ale za trvající povinnosti.

2.5 TASKS (ÚKOLY)

Úkoly jsou různé akce, které musejí být provedeny, aby proběhla aktivita. Související úkoly jsou spojovány dohromady a tvoří tak aktivitu.

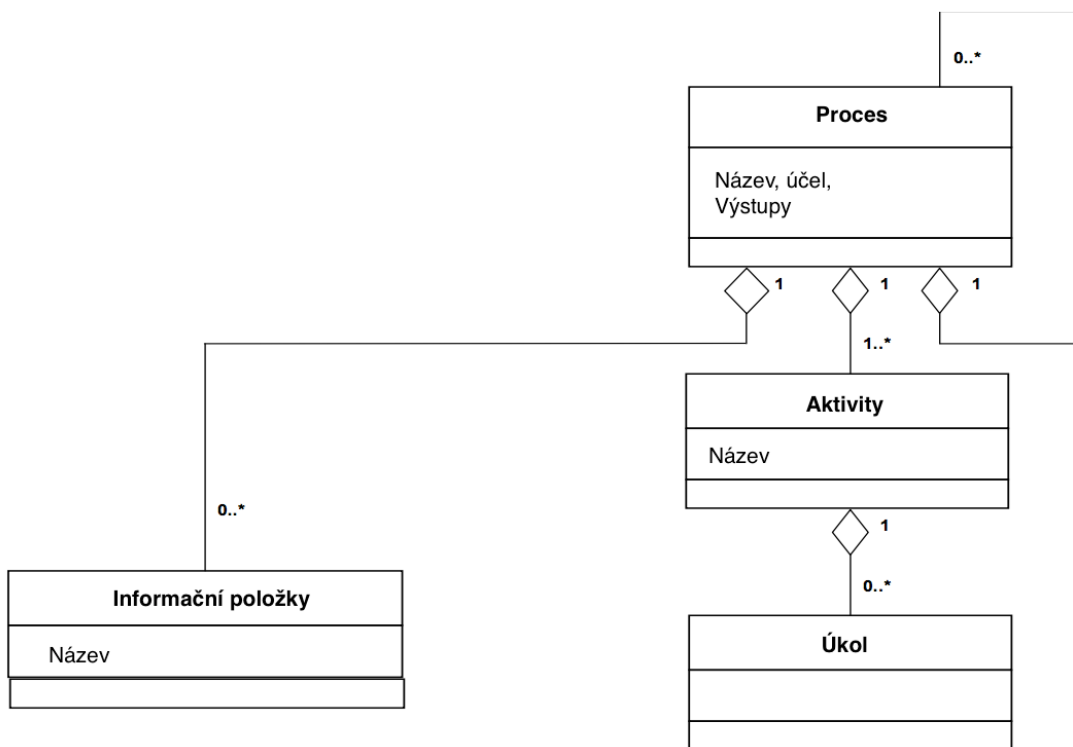
Úkoly jsou vyjádřeny ve formě požadavku, doporučení nebo činnosti.

Platí zde stejná pravidla ohledně plánování času jako u aktivit.

2.6 INFORMATION ITEMS (INFORMAČNÍ POLOŽKY)

Informační položky jsou samostatně identifikovatelné úseky informací používané procesem.

Vstupují a vystupují z procesu. Definuje je popis, specifikace procedůra (způsob nakládání s nimi) a samotný záznam.



Obrázek 1 – Příklad užití elementů v UML reprezentaci procesu [4]

3 COMPOSITION TREE

V předchozích kapitolách bylo popsáno, jaké jsou náležitosti popisu procesu podle normy ISO/IEC TR 24774 a v této kapitole je ukázáno, jak se na základě účelu a výstupů procesu dá namodelovat proces využitím přístupu Composition tree.

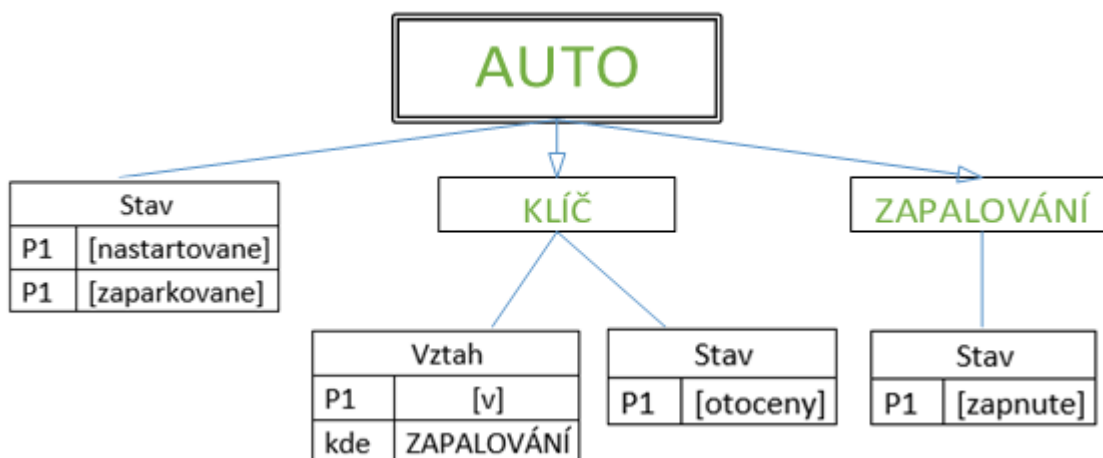
3.1 VYSVĚTLENÍ COMPOSITION TREE

Composition tree byl původně využíván pro popis složení komponentového SW systému. Poskytuje užitečný souhrn informací zahrnujících stavy, atributy a vazby mezi entitami systému. V následující ukázce je nastíněno, jak se vytváří composition tree na základě požadavků na funkcionality a zároveň můžeme vidět základní notaci pro jeho tvorbu.

System AUTO

Požadavky:

- P1: Auto může být nastartované, pouze pokud je zaparkované a klíč je v zapalování a řidič s ním otočí a zapne ho.
- P2: ...PX



Obrázek 2 - Grafické znázornění 1. požadavku na systém využitím přístupu Composition tree (zdroj Martin Baláž, podle [1])

Na Obrázku 1, vidíme ve dvojitém rámečku systém AUTO (může to být zároveň komponenta většího celku, ale pro naši ukázkou je to nejvyšší entita). Z něj potom vychází dva typy vazeb – první, bez šipky, vede k tabulce stavů, ze které můžeme vyčíst, že z prvního požadavku (P1) vyplývají dva stavy – „nastartovane“ a „zaparkovane“. Druhý typ vazby, se šipkou, vede k podřízeným komponentám „KLÍČ“ A „ZAPALOVÁNÍ“, které se stejně jako hlavní entita zapisují velkými písmeny. Dále vidíme, že komponenta KLÍČ má jeden stav – „otoceny“ a má rovněž jeden vztah, „v“, ke komponentě „ZAPALOVÁNÍ“, a ta má taky jeden stav „zapnute“.

Výhody využití CT:

- Všechny požadavky jsou přehledně integrované na jednom místě.
- Jednoduché odhalení vad v požadavcích (nekompletní atd. – každá komponenta alespoň 2 stavy apod.).
- Jasně zobrazení požadavků na každou komponentu – nejsou roztroušené v několika požadavcích.
- Jednoznačnější než přirozený jazyk.
- Konzistentní pojmenování v celém systému.

3.2 VYUŽITÍ PŘÍSTUPU CT K MODELOVÁNÍ SOFTWAREVÝCH PROCESŮ

V následující ukázce využití CT při modelování SW procesů se pracuje s Configuration Management procesem definovaném v normě ISO/IEC 12207:2000, aby byl proces tvorby modelu názornější.

Způsob, kterým se dá sestavit CT z účelu a výstupů, se skládá ze 3 kroků:

1. Přečíst si účel a výstupy a zaznamenat si seznam podstatných jmen a zkratk, což jsou většinou komponenty systému nebo jejich atributy.
2. Vytvoření výchozího CT z účelu – z jeho komponent a stavů.

3. Projít výstupy jeden po druhém, identifikovat komponenty, atributy, stavy a vazby a zakomponovat je do výchozího CT.

Pozn.: v příkladu je zanecháno původní znění, aby překladem netrpěla kvalita a původní myšlenka.

Nejprve tedy projedeme účel a výstupy (podle bodu 1).

Process Name: Software configuration management.

Process Purpose (účel): *The purpose of the Configuration management process is to establish and maintain the integrity of the work products/items of a process or project and make them available to concerned parties.*

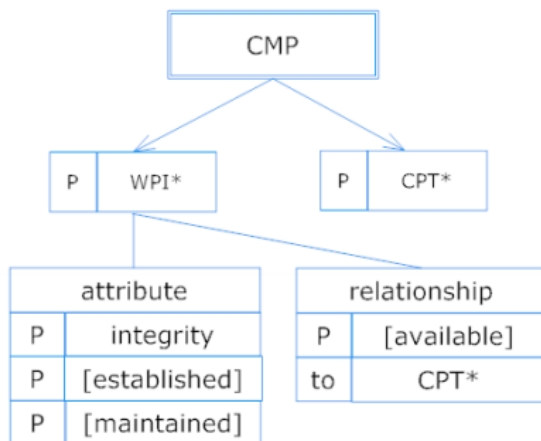
Process Outcomes (výstupy):

1. *a configuration management strategy is developed;*
2. *items generated by the process or project are identified, defined and baselined;*
3. *modifications and releases of the items are controlled;*
4. *modifications and releases are made available to affected parties;*
5. *the status of the items and modifications are recorded and reported;*
6. *the completeness and consistency of the items is ensured; and*
7. *storage, handling and delivery of the items are controlled.*

Identifikované komponenty:

- CMP: Software Configuration Management Process
- WPI: Work product or item
- CPT: Concerned Party
- CMS: Configuration Management Strategy

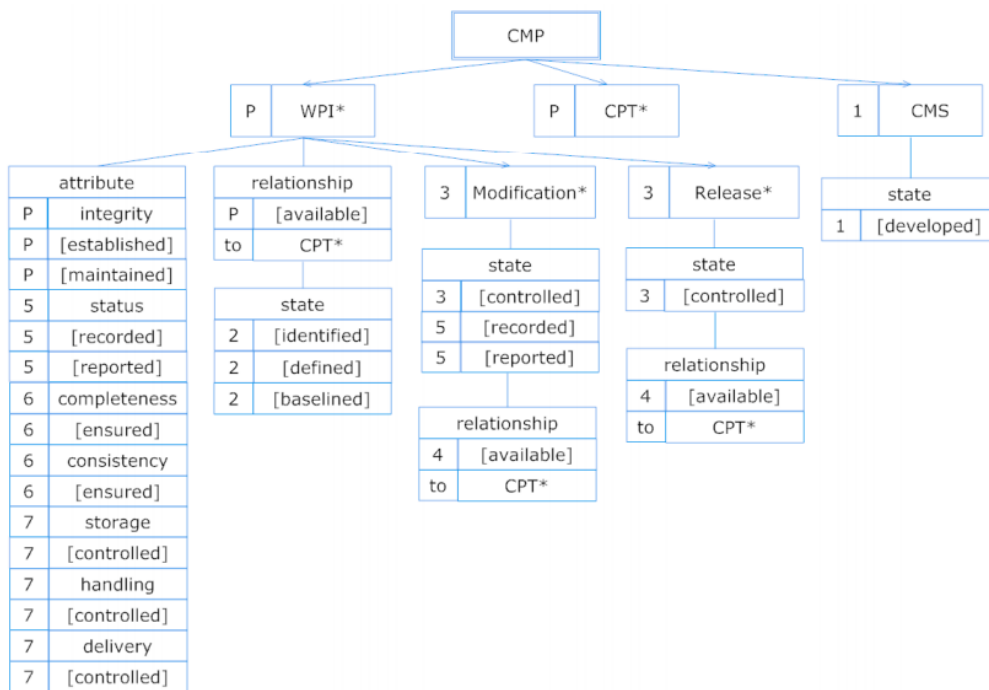
Dalším krokem je sestavení výchozího CT z účelu.



Obrázek 3 – Výchozí CT z účelu Configuration Management procesu [1]

Sestavení výchozího CT z účelu proběhlo obdobně jako v předchozím příkladě sestavení CT z požadavků na AUTO.

A po splnění posledního (3.) kroku, by měl CT vypadat takto:



Obrázek 4 – Kompletní zobrazení Configuration Management procesu využitím přístupu Composition tree [1]

4 MODELÝ PROCESŮ ŽIVOTNÍHO CYKLU SOFTWARE

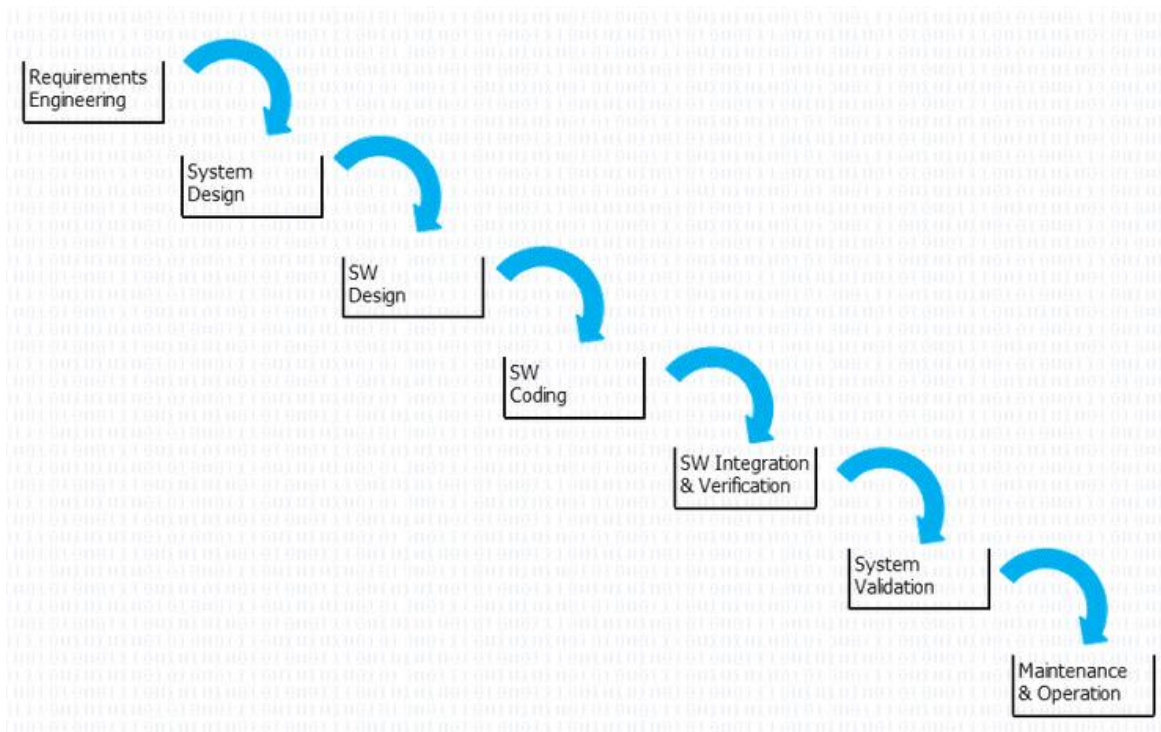
Životním cyklem softwaru se myslí doba od úmyslu jeho vytvoření až po ukončení jeho používání. Modelem rozumíme rámec procesů a aktivit spojených s životním cyklem.

Model životního cyklu software popisuje vzájemné vztahy mezi fázemi životního cyklu softwaru. Každý model obsahuje vlastní metodiku jak zajistit dostatečnou kvalitu produktu. V tomto kontextu často bývá pojem model a metodika zaměňován. Výběr vhodného modelu životního cyklu je klíčový pro úspěch celého projektu. V současné době je modelů životního cyklu software velké množství. Většina z nich ovšem vychází z původních definic vodopádového nebo spirálového modelu. [3]

4.1 VODOPÁDOVÝ MODEL (WATERFALL)

Vodopádový model je nejstarší model životního cyklu software. Jeho pojmenování vychází z přirovnání posloupnosti jednotlivých fází k protékání vody vodopádem, jak je vidět na Obrázku 4. Poprvé jej definoval Winston W. Royce v roce 1970.

- Vodopádový model má 7 základních fází:
 - Analýza požadavků
 - Návrh systému
 - Návrh SW
 - Implementace (Coding)
 - SW integrace a verifikace
 - Systémová validace
 - Provoz a údržba



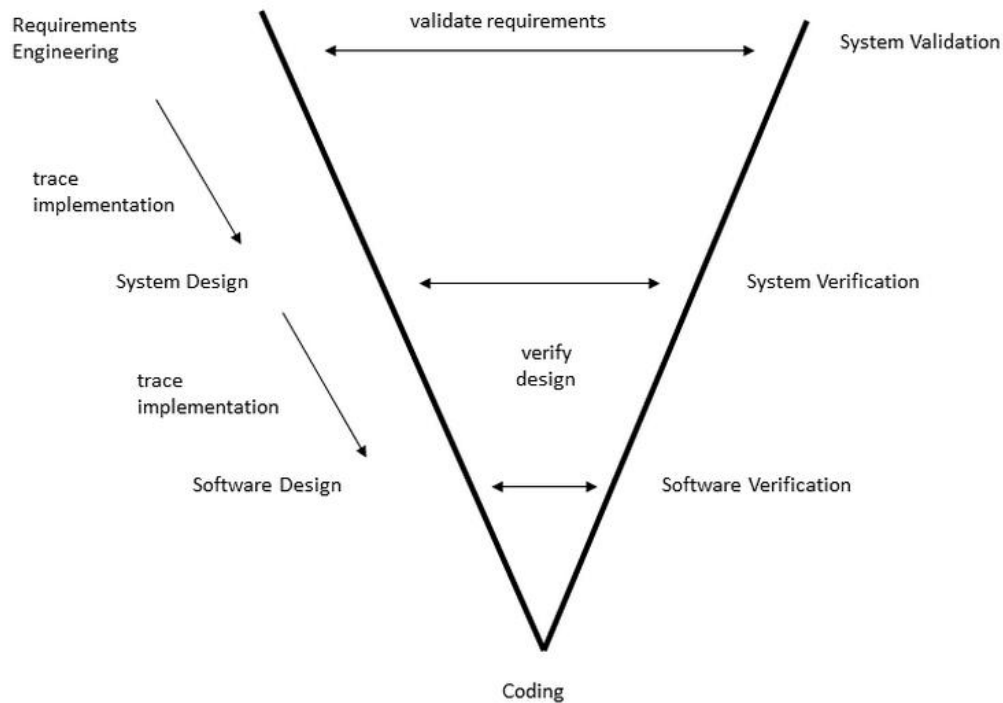
Obrázek 5 – Schéma vodopádového modelu [2]

Základní myšlenka vodopádového modelu, vychází ze sekvenčního přístupu k jednotlivým fázím. Model je charakteristický tím, že vstoupit do další fáze mohou až tehdy, pokud je předchozí kompletně dokončena a uzavřena. Je nutné počátečním fázím věnováno dostatek času (levnější přijít na chybu dříve než později) a iterace je možná pouze v rámci probíhající fáze, maximálně s fází předchozí, takže je důležité na konci každé fáze si být maximálně jisti její validací a kompletností.

Další nevýhody: Nulová možnost reakce na dodatečné požadavky klienta, integrace až po implementaci (časově náročné úpravy a opravy). Jako reakce na nedostatky vznikly modifikace modelu, například V-model.

4.2 V-MODEL

Velmi podobné fáze, ale pro každou fázi návrhu je odvozena odpovídající fáze testování.

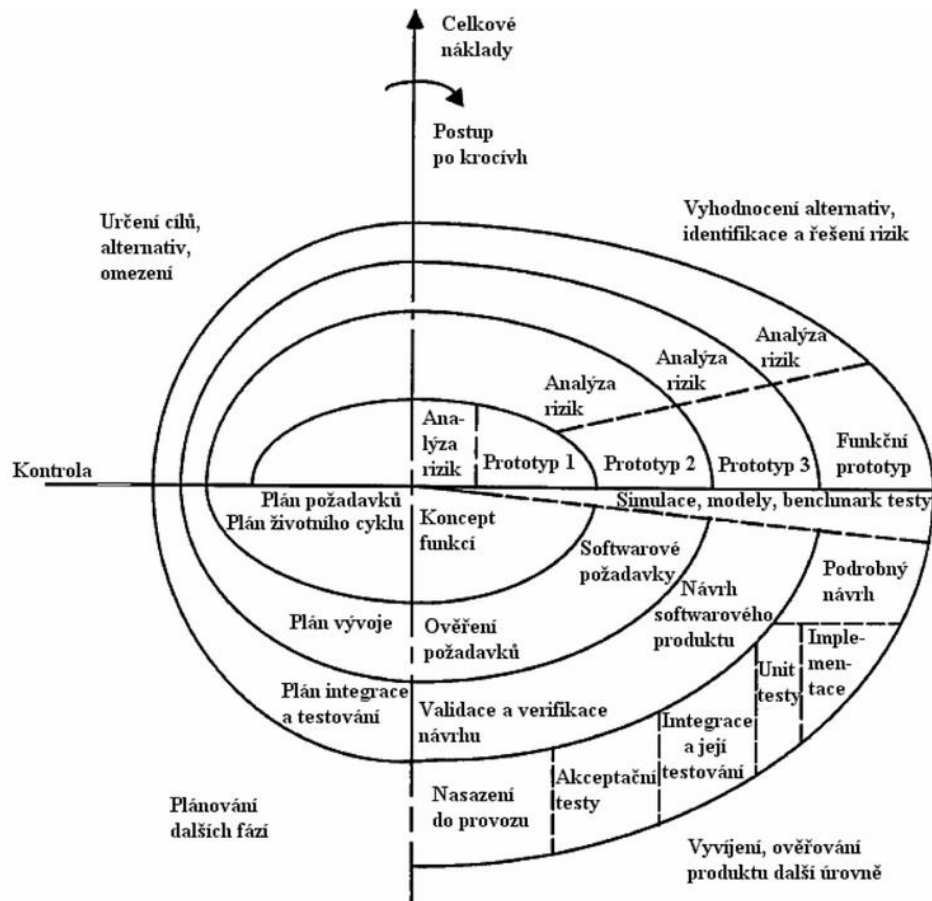


Obrázek 6 – Schéma V-modelu [2]

Požadavek trasovatelnosti shora dolů v levé části modelu – což znamená, že požadavky musejí být sledovány (trace) do návrhu systému, což zaručuje, že budou implementovány kompletně a správně. Pro případ, kde jsou požadavky změněny, byl definován V-cyklus, což je série na sebe navazujících V-modelů, což už je iterativní přístup, na kterém je založen spirálový model.

4.3 SPIRÁLOVÝ MODEL

Spirálový model poprvé definoval Barry Boehm v roce 1986. Tento model velmi dobře pokrývá nedostatky vodopádového modelu.



Obrázek 7 – Schéma spirálového modelu [3]

Model je založen na iterativním přístupu a především zavádí opakovanou analýzu všech rizik. Probíhá v několika krocích, které se neustále opakují, dokud není produkt hotov. Hlavní myšlenkou je zde navazování nových částí na již důkladně prověřený základ. Z počátku se vývoj provádí na základě hrubé specifikace požadavků, v pozdějších fázích je tato specifikace i po konzultaci se zákazníkem postupně upřesňována. Celý životní cyklus podle Spirálového modelu je rozdělen do čtyř hlavních částí:

- 1) Určení cílů, alternativ, omezení
- 2) Vyhodnocení alternativ, identifikace a řešení rizik
- 3) Vývoj a verifikace další úrovně produktu

4) Plánování následujících fází (Plan next phases)

Po každé fázi následuje testování, hodnocení a předání dílčích výsledků. Produkt je tedy testován pravidelně a to již od raných fází – včasné odhalení chyb.

Nevýhody: Nestabilní/špatně naimplementovaný prototyp často bývá finálním produktem, vyžaduje rozsáhlou spolupráci zákazníka -> roste závislost na zákazníkovi, těžko odhadovaná doba trvání projektu.

5 ZÁVĚR

Práce neměla za úkol téma modelů softwarových procesů zcela vyčerpat, ale přes široce definované téma a relativně málo zdrojů se nám podařilo splnit cíle definované v úvodu.

Ukázali jsme jak správně popsat model softwarového procesu dle normy ISO/IEC TR 24774 a předvedli užití elementů na příkladu přímo z této normy. Předvedli jsme také jak modelovat proces s užitím composition tree a shrnuli jednotlivé typy modelů životního cyklu procesu a to Vodopádový model, V-model a Spirálový model.

6 ZDROJE

- [1] MAS, A., A. MESQUIDA, T. ROUT, R.V. O'CONNOR a A. DORLING. Software process improvement and capability determination 12th International Conference, SPICE 2012, Palma, Spain, May 29-31, 2012. Proceedings. Berlin: Springer, 2012. ISBN 978-364-2304-392.
- [2] SW process. TARGET, THE-SOWTWARE-EXPERTS. [online]. 2014 [cit. 2015-03-14]. Dostupné z: http://www.the-software-experts.com/e_dta-sw-process-model.php
- [3] Modely životního cyklu softwaru. TESTOVÁNÍ SOFTWARE. [online]. 2.2.2011 [cit. 2015-05-10]. Dostupné z: <http://testovanisoftwaru.cz/manualni-testovani/modely-zivotniho-cyklu-softwaru/>
- [3] Modely životního cyklu softwaru. TESTOVÁNÍ SOFTWARE. [online]. 2.2.2011 [cit. 2015-05-10]. Dostupné z: <http://testovanisoftwaru.cz/manualni-testovani/modely-zivotniho-cyklu-softwaru/>
- [4] ISO/IEC TR 24774. *Systems and software engineering — Life cycle management — Guidelines for process description*. 2010. WG7.