

ISO/IEC/IEEE 29119 - 1

SEMESTRÁLNÍ PRÁCE KE KURZU 4IT421 ZLEPŠOVÁNÍ PROCESŮ
BUDOVÁNÍ IS

JAKUB ŠÁRKA XSARJ00, VOJTĚH VOTLUČKA XVOTV00

Abstrakt

Cílem této práce je seznámit čtenáře s první částí normy ISO/IEC/IEEE 29119, která se věnuje problematice testování softwaru. Zaměření této části normy je obecné a věcně popisuje problematiku testování. Více do hloubky se věnují další části inkriminované normy. Jako primární zdroj slouží originální norma, a jako sekundární zdroj slouží její český oficiální překlad, jehož kvalita je také následně ohodnocena. Normu lze doporučit k použití, neboť její výstupy a zásady jsou nejen v praxi použitelné, ale i vhodné. Český překlad lze označit za kvalitativně uspokojivý, avšak v některých pasážích slohově kostrbatý.

Klíčová slova

Norma, testování, ISO, IEC, IEEE, organizace, metodika, proces, software, vývoj, ISO/IEC/IEEE 29119

Obsah

1	Úvod	3
2	Norma ISO/IEC/IEEE 29119 - 1	3
2.1	Úvod do testování softwaru	3
2.2	Testování softwaru v kontextu organizace a projektu	3
2.3	Testovací proces	4
2.4	Testovací procesy v životním cyklu softwaru	4
2.5	Testování založené na rizicích	5
2.5.1	Testování založené na rizicích v Procesu testování organizace.....	5
2.5.2	Testování založené na rizicích v Procesech řízení testování	6
2.5.3	Testování založené na rizicích v dynamickém testování	6
2.6	Testovací podproces.....	6
2.6.1	Cíle testování	7
2.6.2	Testovaná položka	7
2.6.3	Testování charakteristik kvality	7
2.6.4	Báze testování	7
2.6.5	Opakované a regresní testování.....	8
2.6.6	Techniky návrhu testování	8
2.7	Praktiky testování.....	8
2.7.1	Testování založené na požadavcích.....	8
2.7.2	Testování založené na modelu	8
2.7.3	Testování založené na matematických postupech.....	8
2.7.4	Testování založené na zkušenostech.....	9
2.7.5	Skriptované a neskriptované testování	9
2.8	Automatizace v testování.....	9
2.9	Správa defektů.....	10
2.10	Osobní názor ke kapitolám 5.1 – 5.4 normy.....	10
2.11	Osobní názor ke kapitolám 5.5 – 5.8 normy.....	10
2.12	Vyjádření k překladu.....	11
3	Závěr	12
4	Literatura	12

1 Úvod

V této práci se budu zabývat normou ISO/IEC/IEEE 29119-1. Tato první část souboru norem se zabývá koncepty a definicemi. Tímto tématem stojí se za to zajímat, jelikož se jedná o standardizovanou normu zabývající se testováním softwaru, které je nyní již nedílnou součástí jakéhokoliv vývoje softwaru.

Za cíl si v práci kladu především představení této normy, její zařazení v rámci celého souboru norem ISO/IEC/IEEE 29119 a vyjádření ke kapitolám normy na základě vlastních zkušeností z oblasti testování softwaru. V neposlední řadě se vyjádřím na základě přečtení originální i přeložené verze normy ke kvalitě překladu.

Hlavním přístupem ke zpracování práce je prostudování především české verze normy a dále za účelem možnosti posouzení kvality překladu i prostudování verze originální. Po představení normy bude následovat zhodnocení zjištěných bodů definovaných normou z pohledu praxe, kde jsem se s problematikou testováním softwaru zabýval.

Vzhledem k tomu že se jedná o pojednání konkrétní normy, nejsou v textu zahrnuty citace.

2 Norma ISO/IEC/IEEE 29119 - 1

2.1 Úvod do testování softwaru

V úvodní kapitole první části normy ISO-29119 se čtenář dozvídá o potřebnosti testování. To je potřebné z několika faktorů, od vyžadování reportu kvality vedoucích osob ve společnosti, přes nutnost odzkoušet funkčnost testovaných položek vůči návrhu, až po nutnost verifikace či validace. Norma také připomíná, že vyhodnocování testovaných položek musí být prováděno během celého životního cyklu vývoje.

Dále norma popisuje obecné informace, proč je testování důležité, jako například proto, že člověk není bezchybný. Následně popisuje, jaký vliv může mít chyba v aplikaci dopad na chod podniku či společnosti. Informace v normě uváděné mají obecnou platnost a domnívám se, že by byli i volně zaměnitelní napříč všemi normami nebo metodikami týkající se testování. Norma například podotýká, že správně získané informace během testování mohou vést nejen k odstranění chyby, ale i k úpravě procesů ve firmě tak, aby se chyby snáze identifikovali či aby se upravili procesy, jenž chyby naopak zakrývají.

V normě se dále uvádějí i stěžejní body testování softwaru. Například to, že testování je proces. Ohledně testovacích procesů se detailněji rozepisuje druhá část normy, stejně tak jako o plánování testování. Dalším bodem je například aplikace testovacích procesů a podprocesů na libovolnou fázi či úroveň testování. Poté přichází na řadu zmínka o statickém a dynamickém testování, o kterým se lze blíže dozvědět také v druhé části normy. A na závěr nastupuje verifikace, tedy ověření uspokojení požadavků a validace, tedy zda je testovaná položka použitelná pro plánované účely. Ovšem verifikace a validace nejsou hlavní součástí normy 29119.

2.2 Testování softwaru v kontextu organizace a projektu

Norma předpokládá, že její použití přinese pozitiva jak po malé vývojářské společnosti, tak pro velké nadnárodní týmy. Malým týmům může přinést řád a postup, jak zvládnout vývojářské procesy z

pohledu testování a velkým týmům naopak pomůže jejich stávající postupy doladit, sjednotit a vylepšit. Norma se k testování staví jako ke strategickému závazku, a doporučuje, aby se do podobné role stavělo i vedení společnosti či týmu. Zároveň však nepředpokládá, že by byla jednoznačně určující autoritou pro způsob testování a vedení testování. Autoři normy si uvědomují, že testování softwaru je natolik specifická činnost, která nejde jednoznačně popsat a tedy i navrhnout. Proto předpokládají, že normy poslouží jako osnova pro zvýšení účinnosti a efektivity testování. Díky normě se také může zavést politika testování a strategie testování do firmy, což přidá testování na potřebné důležitosti a stane se tedy dokonalejším a prospěšnějším.

Testování softwaru je provádění jako proces řízený kontextem. Tedy by mělo být plánováno, monitorováno a řízeno. A to jak v kontextu několikaletých projektů, jednodenních aktivit či standardní údržby systému. Při testování se však také nesmí zapomenout na jiné faktory, jako je například rozpočet, očekávání zákazníka a či rozsah projektu. Další faktory, které ovlivňují testování, mohou být i externího charakteru, jako je například legislativní prostředí, oborové normy a podobně. Také z těchto důvodů neexistuje jednoznačná metoda, která by fungovala ve všech případech. Ovšem průběh i detaily testování se mohou vylepšovat pomocí norem, mezi které patří i zde rozebírané.

Plán testování popisuje celkový postup a strategii testování. Zahrnuje cíle, praktiky, prostředky, časové určení a samozřejmě testovací procesy a podprocesy. Více se však plánu testování věnují další části normy a proto zde nebude podrobněji popsán. V této určité části jsou jen popsány obecné pravdy, které jsou následně hlouběji rozebírány.

2.3 Testovací proces

Norma aplikuje třívrstvý model procesů. Ten se skládá z vrstvy organizační, řídicí a samotného dynamického testování. Samotnému testovacímu procesu se hlouběji věnuje druhá část této normy.

Organizační vrstva spočívá v politice testování a ve strategii testování. Politika testování vyjadřuje očekávaný přístup k testování softwaru od vedení společnosti. Je tedy zaměřena na vyšší management. Udává formu testování v organizaci. Obsah naopak pomáhá plnit strategii testování. V ní jsou obsaženy požadavky a omezení na procesy řízení testování a na dynamické testování. Strategie nemusí být jen jedna, může být vytvořeno i více strategií na míru podle potřeb daného projektu či činnosti. Strategie a politika testování udávají spolu směr, jak má probíhat testování v dané organizaci.

Řídicí vrstva neříká, jak má být prováděno, ale jak má být řízeno. Tedy definuje statické a dynamické testování, role, rozsah a kvalitu. Výstupem této vrstvy je Plán testování projektu a Zpráva o dokončení testů projektu. Během řízení jsou pomocí zpětné vazby prováděny úpravy plánu a testování.

V poslední vrstvě, která je již věnována dynamickému testování, se provádí samotné testování. Norma si dynamické testování představuje jako proces. Na začátku je návrh a implementace testování, po kterém probíhá případné nastavení testovacího prostředí, či rovnou samotné testování. Na základě výsledků se buď vystaví report o incidentu či nikoliv. Zajímavé je, že norma dále nerozpracovává proces po nahlášení incidentu, neboli retestování.

2.4 Testovací procesy v životním cyklu softwaru

Norma stručně popisuje dopady životního cyklu na testování softwaru. Norma uvažuje běžný životní cyklus, tedy od nápadu, vývoje, nasazení, produkci a vyřazení. Sama životní cyklus detailněji nerozebírá, neboť to není obsahem této normy. Zároveň však uvažuje, že každá etapa životního cyklu je podproces, ve kterém se konají jisté činnosti, jako správa požadavků, návrh a programování. A z tohoto důvodu se testování prolíná celým životním cyklem.

Během životního cyklu probíhá i několik podpůrných procesů, jakými jsou zajišťování kvality, správa projektů, správa konfigurace a zdokonalování procesů.

Během zajišťování kvality se pomoci zavedených metod, norem a nástrojů, které se ukázaly jako nejvhodnější pro danou organizaci, kontroluje kvalita a tedy i v důvěra v část výsledku práce. A testování je prostředek, jak toho docílit.

Při správě projektů se testování plánuje a řídí. Tyto aktivity jsou však velmi úzce propojeny s plánováním projektu. Při sestavování plánu samotného projektu se stanoví i rozsah testování. Ovšem na základě výsledku testování se mnohdy musí korigovat i samotný plán celého projektu. Proto se testování musí věnovat patřičná pozornost hned v úvodu projektu.

Správa konfigurace zajišťuje testovací prostředí. Neboli správa zahrnuje jednoznačnou identifikaci, zajištění úložiště, audit uvolňování verzí, řízení změn a reproting. V kontextu testování jsou poté zahrnuty specifikace testování, plán testování, specifikace testování v kontextu organizace či konfigurace testovacích položek.

Testování a zdokonalování procesů jdou ruku v ruce buď tím, že na základě probíhajících testovacích procesů se dá stimul ke zdokonalení jiných procesů anebo tak, že samotné testovací procesy jsou předmětem ke zdokonalení. A samotné zdokonalení se nemusí vztahovat jen na konkrétní projekt, během kterého došlo ke stimulu, ale může ovlivnit procesy v celé organizaci.

2.5 Testování založené na rizicích

Je teoreticky možné otestovat celý software a všechny možné kombinace. Bohužel to však není možné v reálném čase za pomoci lidské síly. Proto se provádí optimální testování na vzorku, který je určen pomocí různých metodik, z nichž jedna je založena na rizicích.

Samotnému řízení rizik se věnuje norma ISO/IEC 16085, proto není detailněji rozebírán. Rizika mohou být identifikována z několika různých úhlů pohledu. První pohled může být neuspokojení regulatorních či zákonných požadavků, dále nevyhovění smluvním závazkem. Další pohled je časové prodlení nebo kvalitativní nedostatky, tedy projektová rizika. A na závěr jsou uvažována i produktová rizika, tedy například nesplnění očekávání zákazníka.

Při testování založené na rizicích jsou tedy rizika nejprve zanalyzována, poté vyhodnocena a vstupují do plánu testování. Během samotného testování jsou rizika korigována.

Testování založené na rizicích se prolíná do několika procesních vrstev, o kterých bude psáno dále.

2.5.1 Testování založené na rizicích v Procesu testování organizace

Politika a strategie testování dané organizace nejen deklarují formu testování v organizaci, ale zároveň určují rizika pro danou organizaci ovlivňující testování. Například automobilový průmysl musí vyhovět kvalitativním požadavkům dané vnější organizací, například státu. K identifikaci rizik v takovém případě pomůže Politika testování.

Naopak strategie testování napomůže jako návod k tomu, jak by testování mělo probíhat, Dále také k řízení rizika selhání a v neposlední řadě by měla pomoci zabránit nesplnění regulačním normám. Například v případě společnosti Volkswagen Strategie testování jejich motorů z pohledu emisí nebyla nastavena tak, aby odpovídala regulačním normám.

2.5.2 Testování založené na rizicích v Procesech řízení testování

Samotné řízení testování probíhá na základě ohodnocení rizik. Hodnocení rizik může vstupovat do plánu testování, neboť je očekávatelné, že riziko s větší vahou dostane větší jak časovou, tak kvalitativní, prioritu, než riziko s menší vahou. Také riziko pomáhá určovat vhodný typ testování dané části. Rizika vtahující se ke komunikaci naznačují, že se bude jednat o integrační testování, ale naopak jiné riziko, například velikost oken a tlačítek ovlivňující uživatelův dojem, nastiňuje spíše testování použitelnosti. Samotné ohodnocení rizik by nemělo probíhat pouze na základě vyjádření jedné strany, ale mělo by se podstoupit vyjádření skrze co možno největšího počtu zúčastněných stran. Pomocí tohoto ohodnocení z několika stran lze dospět k výsledku, že některé riziko je možno považovat za bezpředmětné, i když se zprvu zdálo jako prioritní a naopak.

2.5.3 Testování založené na rizicích v dynamickém testování

Jak již bylo řečeno v přechodí pasáži, hodnocení rizik prostupuje celý procesem testování a tedy zasahuje i do samotného testování. Tedy již ve fázi návrhu testovacího scénáře analytik postupuje podle hodnocení rizik a části, která je ohodnocena větším rizikem, věnuje větší péči a stanoví o širší rozsah testování. Také jednotlivé části se mohou chronologicky testovat dle rizika.

Také podle výsledku testování lze rizika přehodnotit a tedy i upravit plán testování, v případě nutnosti i samotný testovací scénář. Hodnocení rizika lze tedy snížit úspěšným otestováním, či naopak zvýšit neúspěšným otestováním.

2.6 Testovací podproces

Projekt testování je obvykle rozdělen do několika testovacích podprocesů. Způsob dělení závisí na strategii testování projektu. Testovací podproces může být buď spojen s určitou fází životního cyklu softwaru či se zaměřit na zvolenou vlastnost kvality. Podproces se může skládat jak ze statického, tak dynamického testování.

Začátkem testovacího podprocesu je plánování testování. Po celou dobu podprocesu probíhá aktivita monitorování a kontroly testování. Na základě informací z monitorování se může plánování podprocesu v průběhu přehodnotit.

V plánování testování jsou hlavními částmi identifikace cíle testování, rozsahu testování či rizik, která jsou s testovacím podprocesem spojena. Výsledky plánování dále řídí strategii pro podproces, včetně jak statického, tak i dynamického testování.

Popis testovacího podprocesu musí obsahovat informace o tom, které funkce testované položky mají být testovány. To zaručí, že všechny očekávání ohledně rozsahu testování jsou jasné pochopena.

Zpravidla testovací podproces neobsahuje pouze jedno kolo dynamického či statického testování. Pokud se definuje pouze jedno, bývá nutné ho opakovat. V případě dynamického testování se používá opakované testování a regresní testování.

Počet testovacích podprocesů v projektu testování je závislý na strategii testování a fázích životního cyklu. Není ovšem závislý na životním cyklu vývoje.

Řízení výběru aktivit testování a technik návrhu testování je závislé na specifikách testovacího podprojektu. Mezi specifika patří cíl testování, testovaná položka, báze testování či rizika.

2.6.1 Cíle testování

Testy jsou prováděny za účelem dosažení jednoho či více cílů testování. Mezi cíle definované normou ISO/IEC/IEEE 29119 – 1 patří níže uvedené.

- Poskytování informací aktivitě řízení rizik
- Poskytování informací o vlastnostech produktu
- Vyhodnocení splnění očekávání produktu
- Vyhodnocení odstranění defektů
- Vyhodnocení správné implementace
- Vyhodnocení splnění požadavků

Relevantní cíle testování mohou pomoci při určení správných podprocesů.

2.6.2 Testovaná položka

Testovaná položka může být spojená s kódem či dokumentací. Příklady položek spojených s kódem jsou spustitelné softwarové komponenty, podsystém či celý systém. Příklady položek spojených s dokumentací jsou plány, specifikace požadavků, zdrojový kód a další.

2.6.3 Testování charakteristik kvality

Model kvality softwaru je popsán v normě ISO/IEC 25010. Jedná se o těchto 8 charakteristik kvality, které definují atributy kvality testované položky.

- Funkční vhodnost
- Účinnost
- Kompatibilita
- Použitelnost
- Spolehlivost
- Bezpečnost
- Udržovatelnost
- Přenositelnost

K testování charakteristiky kvality může být vyžadováno vytvoření instance podprocesu. Každá z těchto charakteristik se skládá z několika dílčích charakteristik. Všechny charakteristiky kvality ovšem nejsou aplikovatelné na všechny systémy.

2.6.4 Báze testování

Báze testování je definována jako soubor znalostí, na jejichž základě může být testování plánováno, navrženo, vyvozeno, implementováno a řízeno.

Může zahrnovat například očekávanou formu a obsah dokumentace, zkušenosti testera či očekávání implementace komponent.

2.6.5 Opakované a regresní testování

Opakované testování se provádí za účelem vyhodnocení řešení daného incidentu. Dalším typem je regresní testování, které je testování selektivní a provádí se za účelem ověření, zda modifikace v systému nezpůsobily v jiných komponentách nezamýšlené vedlejší efekty.

2.6.6 Techniky návrhu testování

Techniky návrhu testování jsou jak pro statické, tak i dynamické testování. Účelem návrhu je pomoci testerům při nalezení defektů co nejefektivněji. Statické testování toho obvykle dosahuje identifikací zřejmých defektů v popsaných testovaných položkách ve zdrojovém kódu. Dynamické toho dosahuje vynucováním selhání spustitelných testovaných položek.

2.7 Praktiky testování

Praktik testování je mnoho a některé používané dnes jsou uvedeny níže za účelem demonstrace dostupných možností při plánování testování. Technik může být samozřejmě použito víc a navzájem se mohou kombinovat.

2.7.1 Testování založené na požadavcích

Toto testování by mělo zajistit, že požadavky testované položky byly během testování pokryty, aby se zjistilo, zda vyhovuje požadavkům koncových uživatelů. Používá se zde převážně skriptované testování. Během testování v praxi testéři často testují základní požadavky tak, aby ty s největší prioritou byly testovány jako první. Vždy totiž existují časová a rozpočtová rizika a prioritizací se docílí toho, že bude pokryto co nejvíce důležitých požadavků.

2.7.2 Testování založené na modelu

Toto testování odvozuje z modelu informace, jako jsou plány testování, návrh, postupy, vstupy, rizika a další. Výhodou je generování testovacích informací, zlepšené úrovně automatizace testování a brzká identifikace některých typů chyb. Potenciálně velké množství testovacích případů tak může být pokryto v relativně krátkém časovém období. Grafické UML notace jsou často používány jako základ těchto modelů. Další výhodou tohoto testování je i to, že modifikace modelu je jednodušší než aktualizace testovacích skriptů (následná generace a exekuce aktualizovaných testů s sebou nese relativně nízké náklady).

2.7.3 Testování založené na matematických postupech

Testování založené na matematických postupech může být použito k plánování, návrhu, výběru dat či nastavení vstupních podmínek. Zmenšují také problém lidské předpojatosti při nastavení priorit testovacích případů. Tyto postupy užívají několik technik, například kombinační testování, fuzzy testování, shlukové vzorkování apod. Vzhledem k velkému množství vstupů jsou pro tento typ testování obvykle potřeba automatizované nástroje.

2.7.4 Testování založené na zkušenostech

Testování založené na zkušenostech se zakládá na předchozích zkušenostech s testováním, znalosti konkrétního softwaru, doménových znalostech či metrikách z předešlých projektů. V normě ISO/IEC/IEEE 29119-4 je tento typ testování označen jako „Odhadování chyb“. Patří sem například průzkumné testování či Ad hoc testování. Při těchto typech testování se nespolehá na dokumentaci, ale spíše na intuici či zvědavost testera. Proto je tento typ převážně neskriptovaný.

2.7.5 Skriptované a neskriptované testování

Testovací techniky, jak bylo výše zmíněno, používají jak skriptované, tak neskriptované testování. Oba typy mají své výhody i nevýhody. V praxi se tyto typy často kombinují. Primárním hlediskem při rozhodování ohledně typu testování je profil rizika testované položky. Neskriptované testování by tak mohlo být použito pro položky s nízkým rizikem.

Skriptované testování má řadu výhod, mezi něž patří opakovatelnost testů, jejich dosledovatelnost zpět k požadavkům či uchování testovacích případů v podobě znovupoužitelných artefaktů. Nevýhody jsou především časová náročnost a nákladnost, menší schopnost adaptace během testování či možnost ztráty pozornosti testerů vzhledem k tomu, že je méně podnětné a nevyžaduje analytickou práci.

Mezi výhody neskriptovaného testování patří to, že tester není omezen skriptem a může se inspirovat nápady během exekuce v reálném čase. Dále může testovanou položku přezkoumat rychle a přizpůsobit chování systému v reálném čase. Mezi nevýhody patří to, že testy nejsou většinou opakovatelné. Pro provádění neskriptovaného testování jsou také potřeba většinou zkušenější testéři, kteří vědí, na co se zaměřit. V neposlední řadě je nevýhodou to, že testy neposkytují záznam o tom, které testování bylo dokončeno a může tak být obtížné měřit tento proces.

2.8 Automatizace v testování

Automatizace testování může být použita k podpoře testovacích praktik výše zmíněných. Kupříkladu testování založené na modelu je téměř vždy založeno na užití nástrojů pro automatizované provádění. Automatizace testování vyžaduje použití testovacích nástrojů. Nástrojů existuje velká řada, a to jak komerčních, tak i nástrojů bezplatných. Oblasti, které mohou tyto nástroje pokrýt, jsou například tyto:

- Správa testovacích případů
- Monitorování a kontrola testování
- Generování testovacích dat
- Statická analýza
- Generace a exekuce testovacích případů
- Implementace a údržba testovacího prostředí

2.9 Správa defektů

Za správu defektů je zpravidla odpovědný vedoucí testování projektu. Správa defektů je pokryta především v druhé normě souboru norem 29119. Tato správa pomáhá testerům při vyšetřování a dokumentaci zpráv o incidentech.

2.10 Osobní názor ke kapitolám 5.1 – 5.4 normy

V této kapitole budou popsány první čtyři kapitoly první části normy z pohledu testera a test analytika v nejmenované bance, tedy z pohledu praxe. V další kapitole budou popsány ostatní pasáže normy opět z jiného pohledu.

V kapitole 5.1 Úvod do testování softwaru se norma věnuje obecnému popisu testování a proč je potřebné. S jejími závěry nelze nesouhlasit, poněvadž testování softwaru je opravdu potřebné. Osobně se tážím, zda je taková pasáž v normě opravdu potřebná, neboť organizaci tato část nepředá žádnou podstatnou informaci. Očekávám, že každá společnost, která se aktivně zajímá o tuto normu, problematiku testování jistě zná. Jediným přínosem je aspoň vyjádření definic výrazů používaných v normě, i když ke stejnému účelu slouží slovníček v normě také obsažený.

V následující části se norma věnuje testování z pohledu organizace a zasazuje jej do kontextu dané organizace. Její vysvětlení, jak má být testování řízeno a aplikováno je relativně obsáhlé. A opět nelze s normou nesouhlasit. Jen jsem v praxi nezažil politiku či strategii testování. V nejmenované bance, kde pracuji, bych to spíše nazval kulturou testování. Je pravda, že tyto procesy nejsou v bance jasně definovány a jsou uskutečňovány mnohdy nahodile. Proto se domnívám, že touto částí by se mohla banka inspirovat a dále se podle ní rozvíjet. Jedná se tedy o první část, kde shledávám přínos v praxi. Bohužel mám však na druhou stranu dojem, že takto by se mohla vyjadřovat jakákoliv jiná norma, protože jazyk je stále velmi obecný.

Dále norma rozvádí testování v životním cyklu softwaru. Její stanoviska jsou opět velmi obecná. Zajímavý je přechod od vysvětlování životního cyklu softwaru a testování během něj k postupně vysvětlování řízení testování. Nevím, zda autoři normy záměrně takto strukturovali kapitolu, ale nabyl jsem dojmu, že tato a předchozí kapitola obsahově velmi splývají a proto bych doporučil jasnější strukturalizaci. Dlužno uznat, že zde se norma rozpráví více do hloubku a obsah kapitole je již v praxi aplikovatelný, například ohledně propojení plánu projektu a plánu testování. Přijde mi, že v praxi se tomuto propojení bohužel nevěnuje příliš prostoru.

Ve čtvrté kapitole norma popisuje testování založené na rizicích, jako jednu z metod, jak plánovat testování. Osobně s tímto přístupem naprosto souhlasím. Přijde mi, že v praxi se prosazuje přirozeně. Chybí mi však rozvinutí jiných přístupů, protože se nedomnívám, že testování založené na rizicích je v praxi vždy použitelné a v normě by tedy měla být zmíněn i alternativní přístup. Bohužel, obsah této kapitoly shledávám jako velmi popisný a nedosahující optimální obsahové hloubky.

2.11 Osobní názor ke kapitolám 5.5 – 5.8 normy

V této kapitole bych rád vyjádřil názor ke kapitolám 5.5 – 5.8 rozebraných v této práci. Jmenovitě se jedná o kapitoly Testovací podproces, Praktiky testování, Automatizace testování a Správa defektů.

Ačkoliv se v oblasti testování softwaru již nějaký rok pohybují, první ze zmíněných kapitol, tedy Testovací podproces, obsahovala některé informace a terminologii, s kterými jsem se prozatím

nesetkal. Kupříkladu již samotný název kapitoly je něco, co jsem v praxi nikdy nezaslechl. Samotné vysvětlení podprocesu je sice popsáno poměrně jasně, ovšem zde se jednalo o novinky, alespoň co se terminologie týká. Věřím, že informace z této kapitoly mohou být pro testery užitečné, ovšem přímo zde zmíněné pojmy se do praxe pravděpodobně příliš nezavedly.

Na druhou stranu další kapitola, tj. Praktiky testování, již pojednává o něčem, s čímž jsem se i v praxi již setkal a zde bych řekl, že norma pokrývá ty základní praktiky, s kterými se tester může setkat. Zajisté jich může být více, ale jak je v normě zmíněno, jedná se pouze o popis některých praktik. Přesto jsou zde zmíněny i například praktiky založené na matematických postupech, které nejsou natolik rozšířené, jako například testování založené na požadavcích. Dále i popis výhod a nevýhod skriptovaného i neskriptovaného testování obsahuje věci, které jsou snadno ověřitelné i v praxi a jejich výčet může pomoci v praxi ke správné volbě typu testování.

Co se kapitol Automatizace testování a Správa defektů týče, jedná se o kapitoly krátké a rozšiřované v jiných normách. I přesto například výčet oblastí pokrytých testovacími nástroji obsahuje užitečné příklady, kdy může přijít vhod. Správa defektů zde je pravděpodobně jen do počtu, jelikož v jednom odstavci, který ji tvoří, je tvořena především referencemi na jiné normy, které poskytnou podrobnější informace.

2.12 Vyjádření k překladu

Ačkoliv v této práci bylo čerpáno z české verze normy ISO/IEC/IEEE 29119-1, přečetl jsem i její anglický originál. I když mám nějaké překlady také za sebou, nemohu se považovat za odborníka, a tak se zde pokusím vyjádřit svůj názor z pohledu čtenáře normy.

Česká verze normy se mi obecně jeví jako poměrně povedený překlad, ačkoliv na některých místech může být slovosled trochu zvláštní a pravděpodobně spojen s přesným překládáním z originálu. Také obecně v celém textu by srozumitelnosti normy pravděpodobně prospělo místo souvětí, které někdy zabírají celý odstavec, například rozdělení na několik dílčích vět.

Co se faktického překladu nemám příliš výhrad. Ačkoliv jsem nešel s překlady slovo po slovu, povedlo se mi zachytit jednu faktickou nepřesnost. Konkrétně v kapitole Testovací podproces, kde se v celém textu řeší testovací podproces, ale v posledním odstavci úvodu ke kapitole je zmíněno „...rizika jsou specifická pro testovací podprojekt...“. Zde v originále opět zmiňují podproces, ale v české verzi z nějakého důvodu vystupuje podprojekt.

Mimo tuto faktickou nepřesnost jsem natrefil kupříkladu na překlep „Opakvné a regresní testování“. Toto není již věc překladu, ale spíše korektury.

Dalšími detaily, které se vyskytují v textu, jsou místy nepřesné skloňování, které vede k tomu, že informace nemůže čtenář zcela přesně pojmout. V podkapitole Báze testování je kupříkladu zmíněno, že „Příklady báze testování mohou zahrnovat: - Očekávání formu a obsahu dokumentace...“. Toto se mi z českého překladu nepovedlo pochopit. Z originálu to již bylo jasné.

Nebudu se ovšem pouštět do podrobného „rozpitvávání“ vynechaných čárek apod., jelikož to nevypovídá již tolik o kvalitě překladu, který mimo pár ukázek nepřesných detailů, je povedený a věřím, že čtenář je schopen naprostou většinu normy pochopit i z české verze normy.

3 Závěr

Závěrem lze říci, že první část normy ISO/IEC/IEEE 29119 obsahuje potřebné úvodní informace, avšak potřebná hloubka a přínos se skrývá v následujících částech. První část je užitečná pro stanovení definic, charakteristik výrazů a spojení používaných v normě. Norma je svojí hloubkou dostatečná pro pochopení souvislostí, ale nedostatečná pro použití v praxi. Očekávám však, že následující části normy jsou již dostatečné a přínosné.

Její stanoviska lze uplatnit v praxi, a také se v praxi uplatňují. Domnívám se však, že je to způsobené z nepodstatné míry obecností pravd v normě používaných. Důležité jsou souvislosti, na které norma upozorňuje. Použitelnost normy v praxi lze tedy považovat za dostatečnou.

Pro české organizace je přínosné, že existuje i český překlad normy, který lze považovat za kvalitní pro profesionální sféru. Ovšem jak norma, tak překlad obsahují i spojení v praxi nepoužívané a neznámé, což může zapříčinit její odmítání mezi pracovníky.

Norma je jistě v dnešní době agilního vývoje softwaru potřebná, a proto za ní budme rádi, protože může napomoci sjednotit a zlepšit principy testování používané v praxi.

4 Literatura

Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2014. *Softwarové a systémové inženýrství – Testování softwaru – Koncepty a definice* [online]. [cit. 2015-12-12]. Dostupné z: https://vse-my.sharepoint.com/personal/buchalc_vse_cz/Documents/Mater%C3%A1ly%20pro%20IT421/ISO29119.ZIP

ISO/IEC/IEEE, 2013. *Software and systems engineering – Software testing – Part 1: Concepts and definitions* [online]. [cit. 2015-12-12]. Dostupné z: https://vse-my.sharepoint.com/personal/buchalc_vse_cz/Documents/Mater%C3%A1ly%20pro%20IT421/ISO29119.ZIP