



Certifikovaný tester

Učebná osnova pre základný stupeň

Verzia 2011 SK Beta 2



Upozornenie k autorským právam

Tento dokument môže byť kopirovaný v celom svojom rozsahu, alebo časti, pokiaľ je uvedený zdroj.

Copyright © Medzinárodný výbor pre kvalifikácie testovania softvéru - International Software Testing Qualifications Board (v ďalšom texte označovaný ISTQB®).

ISTQB je registrovanou značkou Medzinárodný výbor pre kvalifikácie testovania softvéru - International Software Testing Qualifications Board.

Copyright © 2011, autori aktualizovanej verzie (Thomas Müller (predseda), Debra Friedenberg a Pracovná skupina ISTQB pre základný stupeň).

Copyright © 2010, autori aktualizovanej verzie (Thomas Müller (predseda), Armin Beer, Martin Klonek, Rahul Verma).

Copyright © 2007, autori aktualizovanej verzie (Thomas Müller (predseda), Rex Black, Sigrid Eldh, Dorothy Graham, Klaus Olsen, Maaret Pyhäjärvi, Geoff Thompson a Erik van Veenendaal).

Copyright © 2005, autori (Thomas Müller (predseda), Rex Black, Sigrid Eldh, Dorothy Graham, Klaus Olsen, Maaret Pyhäjärvi, Geoff Thompson a Erik van Veenendaal).

Všetky práva vyhradené.

Autori týmto prevádzajú autorské právo na Medzinárodný výbor pre kvalifikácie testovania softvéru (v ďalšom texte označovaný ISTQB). Autori (ako súčasní držiteľia autorského práva) a ISTQB (ako budúci držiteľ autorského práva) sa dohodli na nasledujúcich podmienkach užívania:

- 1) Akákoľvek osoba alebo tréningová spoločnosť môže použiť túto učebnú osnovu ako základ pre tréningový kurz v prípade, že autori a ISTQB sú potvrdení ako zdroj a vlastníci práv tejto učebnej osnovy. Zároveň musí byť zabezpečené, že akákoľvek propagácia takéhoto tréningového programu môže spomenúť túto učebnú osnovu len v prípade predloženia oficiálnej akreditácie tréningových materiálov uznaným lokálnym výborom ISTQB.
- 2) Akákoľvek osoba alebo skupina môže použiť túto učebnú osnovu ako základ pre články, knihy alebo iné druhotné písomné záznamy v prípade, že autor a ISTQB sú potvrdení ako zdroj a vlastníci práv tejto učebnej osnovy.
- 3) Ktorýkoľvek lokálny výbor uznaný ISTQB môže preložiť túto učebnú osnovu a licencovať učebnú osnovu (alebo jej preklad) iným stranám.

História zmien slovenskej verzie

Verzia	Dátum	Poznámky
ISTQB 2011 SK Beta 2	9.10.2011	Certifikovaný tester Základný stupeň – slovenský preklad Verzia Beta 2.
ISTQB 2011 SK Beta 1	30.9.2011	Certifikovaný tester Základný stupeň – slovenský preklad Verzia Beta 1.
ISTQB 2007 SK Beta 2	10.3.2008	Certifikovaný tester Základný stupeň – slovenský preklad Verzia Beta 2.
ISTQB 2007 SK Beta 1	1.2.2008	Certifikovaný tester Základný stupeň – slovenský preklad Verzia Beta 1.

História zmien angického originálu

Verzia	Dátum	Poznámky
ISTQB 2011	30.3.2011	Certified Tester Foundation Level Syllabus Maintenance Release – see Appendix E – Release Notes Syllabus 2010
ISTQB 2010	30.3.2010	Certified Tester Foundation Level Syllabus Maintenance Release – see Appendix E – Release Notes Syllabus 2010
ISTQB 2007	1.5.2007	Certified Tester Foundation Level Syllabus Maintenance Release – see Appendix E – Release Notes Syllabus 2007
ISTQB 2005	1.7.2005	Certified Tester Foundation Level Syllabus
ASQF V2.2	Júl 2003	ASQF Syllabus Foundation Level Version 2.2 “Lehrplan „Grundlagen des Software-testens“
ISEB V2.0	25. február 1999	ISEB Software Testing Foundation Syllabus V2.0 25 February 1999

Obsah

Podakovanie.....	7
Úvod k učebnej osnove	8
Účel dokumentu	8
Základný stupeň Certifikovaný tester v testovaní softvéru	8
Študijné ciele / kognitívna úroveň znalosti.....	8
Skúška	8
Akreditácia	8
Úroveň detailu.....	9
Ako je táto učebná osnova usporiadaná	9
1. Základy testovania (Z2)	10
1.1 <i>Prečo je potrebné testovanie (Z2)</i>	11
1.1.1 Súvislosti v softvérových systémoch (Z1).....	11
1.1.2 Príčiny softvérových defektov (Z2)	11
1.1.3 Úloha testovania v softvérovom vývoji, údržbe a prevádzke (Z2).....	11
1.1.4 Testovanie a kvalita (Z2)	11
1.1.5 Kedy je testovanie dostatočné? (Z2)	12
1.2 <i>Čo je testovanie? (Z2)</i>	13
1.3 <i>Sedem princípov testovania (Z2)</i>	14
1.4 <i>Základný testovací proces (Z1)</i>	15
1.4.1 Plánovanie a riadenie testovania (Z1)	15
1.4.2 Analýza a návrh testovania (Z1)	15
1.4.3 Implementácia a vykonanie testovania (Z1)	16
1.4.4 Vyhodnotenie výstupných kritérií a reportovanie (Z1)	16
1.4.5 Aktivity uzatvorenia testu (Z1)	16
1.5 <i>Psychológia testovania (Z2)</i>	18
1.6 <i>Etický kódex</i>	20
2. Testovanie v životnom cykle softvéru (Z2)	21
2.1 <i>Modely softvérového vývoja (Z2)</i>	22
2.1.1 V-model (sekvenčný vývojový model) (Z2).....	22
2.1.2 Iteratívno-inkrementálne vývojové modely (Z2).....	22
2.1.3 Testovanie v modeli životného cyklu (Z2)	22
2.2 <i>Úrovne testovania (Z2)</i>	24
2.2.1 Testovanie komponentov (Z2)	24
2.2.2 Integračné testovanie (Z2)	25
2.2.3 Systémové testovanie (Z2)	25
2.2.4 Akceptačné testovanie (Z2)	26
2.3 <i>Typy testov (Z2)</i>	28
2.3.1 Testovanie funkcionality (funkcionálne testovanie) (Z2)	28
2.3.2 Testovanie nefunkcionálnych softvérových charakteristík (nefunkcionálne testovanie) (Z2)	28
2.3.3 Testovanie štruktúry/architektúry softvéru (štrukturálne testovanie) (Z2)	29
2.3.4 Testovanie súvisiace so zmenami: retestovanie a regresné testovanie (Z2).....	29
2.4 <i>Testovanie údržby (Z2)</i>	30
3. Statické techniky (Z2)	31
3.1 <i>Statické techniky a proces testovania (Z2)</i>	32
3.2 <i>Revízny proces (Z2)</i>	33
3.2.1 Aktivity formálnej revízie (Z1)	33
3.2.2 Roly a zodpovednosti (Z1).....	33
3.2.3 Typy revízií (Z2).....	34
3.2.4 Faktory úspechu pre revízie (Z2).....	35
3.3 <i>Statická analýza s použitím nástrojov (Z2)</i>	36
4. Techniky tvorby testov (Z4)	37

4.1	<i>Proces vývoja testu (Z3)</i>	39
4.2	<i>Kategórie techník návrhu testov (Z2)</i>	40
4.3	<i>Techniky založené na špecifikácii alebo techniky čiernej skrinky (Z3)</i>	41
4.3.1	Rozdelenie ekvivalencie (Z3).....	41
4.3.2	Analýza hraničných hodnôt (Z3).....	41
4.3.3	Testovanie rozhodovacích tabuliek (Z3).....	41
4.3.4	Testovanie prechodu stavov (Z3)	42
4.3.5	Testovanie prípadov použitia (Z2)	42
4.4	<i>Techniky založené na štruktúre alebo techniky bielej skrinky (Z4)</i>	43
4.4.1	Testovanie a pokrytie príkazov (Z4)	43
4.4.2	Testovanie a pokrytie rozhodovaní (Z3)	43
4.4.3	Ďalšie techniky založené na štruktúrach (Z1).....	43
4.5	<i>Techniky založené na skúsenosti (Z2)</i>	45
4.6	<i>Výber testovacích techník (Z2)</i>	46
5.	<i>Manažment testovania (Z3)</i>	47
5.1	<i>Organizácia testovania (Z2)</i>	49
5.1.1	Organizácia testovania a nezávislosť (Z2)	49
5.1.2	Úlohy vedúceho testovania a testera (Z1).....	49
5.2	<i>Plánovanie a odhadovanie testovania (Z3)</i>	51
5.2.1	Plánovanie testovania (Z2).....	51
5.2.2	Aktivity plánovania testovania (Z3).....	51
5.2.3	Vstupné kritériá (K2).....	51
5.2.4	Výstupné kritériá (Z2)	52
5.2.5	Odhadovanie testovania (Z2)	52
5.2.6	Stratégia testovania, prístup k testovaniu (Z2).....	52
5.3	<i>Sledovanie a riadenie postupu testovania (Z2)</i>	54
5.3.1	Sledovanie postupu testovania (Z1)	54
5.3.2	Reportovanie z testovania (Z2).....	54
5.3.3	Riadenie testovania (Z2).....	54
5.4	<i>Konfiguračný manažment (Z2)</i>	56
5.5	<i>Riziko a testovanie (Z2)</i>	57
5.5.1	Projektové riziká (Z2).....	57
5.5.2	Produktové riziká (Z2).....	57
5.6	<i>Manažment incidentov (Z3)</i>	59
6.	<i>Podporné nástroje pre testovanie (Z2)</i>	61
6.1	<i>Typy testovacích nástrojov (Z2)</i>	62
6.1.1	Podporné nástroje pre testovanie	62
6.1.2	Klasifikácia testovacích nástrojov (Z2)	62
6.1.3	Podporné nástroje pre manažment testovania a testov (Z1)	63
6.1.4	Podporné nástroje pre statické testovanie (Z1).....	63
6.1.5	Podporné nástroje pre špecifikáciu testov (Z1)	64
6.1.6	Podporné nástroje pre vykonanie a zaznamenávanie testov (Z1)	64
6.1.7	Podporné nástroje pre výkon a monitorovanie (Z1)	64
6.1.8	Podporé nástroje pre špecifické potreby testovania (Z1)	65
6.2	<i>Efektívne použitie nástrojov: možné výhody a riziká (Z2)</i>	66
6.2.1	Možné výhody a riziká nástroja pre podporu testovania (pre všetky nástroje) (Z2).....	66
6.2.2	Osobitné úvahy ku niektorým typom nástrojov (Z1)	66
6.3	<i>Zavedenie nástroja v organizácii (Z1)</i>	68
7.	Referencie.....	69
8.	Príloha A – Pozadie učebnej osnovy	71
9.	Príloha B – Študijné ciele / kognitívna úroveň znalostí.....	73
10.	Príloha C – Pravidlá používané pre základnú učebnú osnovu ISTQB.....	75
10.1.1	Všeobecné pravidlá	75
10.1.2	Aktuálny obsah	75

10.1.3	Študijné ciele	75
10.1.4	Celková štruktúra.....	75
11.	Príloha D – Upozornenie poskytovateľom školení	77
12.	Príloha E – Poznámky k vydaným verziám	78
13.	Index	80

PodĎakovanie

Pracovná skupina “Základný stupeň” Medzinárodného výboru pre kvalifikácie testovania softvéru (International Software Testing Qualifications Board Working Group Foundation Level), edícia 2011:

Thomas Müller (predseda), Debra Friedenberg. Kľúčový tím ďakuje revíznemu tímu (Dan Almog, Armin Beer, Rex Black, Julie Gardiner, Judy McKay, Tuula Pääkkönen, Eric Riou du Cosquer, Hans Schaefer, Stephanie Ulrich, Erik van Veendendaal) a všetkým lokálnym výborom za návrhy k súčasnej učebnej osnove.

Pracovná skupina “Základný stupeň” Medzinárodného výboru pre kvalifikácie testovania softvéru (International Software Testing Qualifications Board Working Group Foundation Level), edícia 2010:

Thomas Müller (predseda), Rahul Verma, Martin Klonk a Armin Beer. Kľúčový tím ďakuje revíznemu tímu (Rex Black, Mette Bruhn-Pederson, Debra Friedenberg, Klaus Olsen, Judy McKay, Tuula Pääkkönen, Meile Posthuma, Hans Schaefer, Stephanie Ulrich, Pete Williams, Erik van Veendendaal) a všetkým lokálnym výborom za ich návrhy.

Pracovná skupina “Základný stupeň” Medzinárodného výboru pre kvalifikácie testovania softvéru (International Software Testing Qualifications Board Working Group Foundation Level), edícia 2007:

Thomas Müller (predseda), Dorothy Graham, Debra Friedenberg a Erik van Veendendaal. Kľúčový tím ďakuje revíznemu tímu (Hans Schaefer, Stephanie Ulrich, Meile Posthuma, Anders Pettersson a Wonil Kwon) a všetkým lokálnym výborom za ich návrhy.

Pracovná skupina “Základný stupeň” Medzinárodného výboru pre kvalifikácie testovania softvéru (International Software Testing Qualifications Board Working Group Foundation Level), edícia 2005:

Thomas Müller (predseda), Rex Black, Sigrid Eldh, Dorothy Graham, Klaus Olsen, Maaret Pyhäjärvi, Geoff Thompson a Erik van Veenedendaal. Kľúčový tím ďakuje revíznemu tímu a všetkým lokálnym výborom za ich návrhy.

Preklad do slovenského jazyka - Czech and Slovak Testing Board (CaSTB):

2011: Róbert Dankanin, Marek Majerník, Ľuboš Práznovský.

Revízia prekladu do slovenského jazyka: Daniela Čuvarská, Marcel Veselka.

2008: Daniela Čuvarská, Róbert Dankanin, Karol Frühauf, Marek Majerník, Ľuboš Práznovský.

Revízia prekladu do slovenského jazyka: Roman Jurkech.

Úvod k učebnej osnove

Účel dokumentu

Táto učebná osnova tvorí základ pre medzinárodnú kvalifikáciu softvérového testovania pre základný stupeň. Medzinárodný výbor pre kvalifikácie softvérového testovania (International Software Testing Qualifications Board, ďalej už len ISTQB®) ich poskytuje lokálnym výborom, ktoré akreditujú poskytovateľov školení a odvodzujú skúšobné otázky v ich materinskom jazyku. Poskytovatelia školení určujú vhodné výukové metódy a vytvárajú učebné materiály pre akreditáciu. Táto učebná osnova pomáha kandidátom v príprave na skúšku.

Informácie o histórii a pozadí učebnej osnovy sa nachádzajú v Prílohe A.

Základný stupeň Certifikovaný tester v testovaní softvéru

Kvalifikácia "Základný stupeň" je určená pre kohokoľvek, kto je určitým spôsobom zapojený do testovania softvéru. To znamená roly ako tester, test analytik, test inžinier, test konzultant, test manažér, užívateľský akceptačný tester a softvérový vývojár. Kvalifikácia základného stupňa je tiež vhodná pre každého, kto chce v základoch porozumieť testovaniu softvéru, napr. pre projektových manažérov, manažérov kvality, manažérov vývoja softvéru, biznis analytikov, riaditeľov IT a konzultantov v manažmente. Držitelia certifikátu základného stupňa budú môcť pokračovať vo vyšších úrovniach kvalifikácie.

Študijné ciele / kognitívna úroveň znalosti

Študijné ciele sú označené pre každú časť tejto učebnej osnovy a klasifikované nasledovne:

- Z1: zapamätať si;
- Z2: pochopiť;
- Z3: aplikovať;
- Z4: analyzovať.

Bližšie detaily a príklady študijných cieľov sú uvedené v Prílohe B.

Je potrebné zapamätať si (Z1) všetky termíny uvedené v časti "Základné výrazy" priamo pod názvom kapitoly, hoci nie sú explicitne uvedené v cieľoch učenia sa. Študijné ciele sú v osnovách tiež označované ako ŠC.

Skúška

Certifikačná skúška pre základný stupeň bude založená na tejto učebnej osnove. Odpovede na skúšobné otázky môžu vyžadovať použitie viac ako jednej časti osnov. Skúšať je možné všetky časti učebnej osnovy.

Forma skúšania je výber z viacerých možností.

Skúšky môžu byť vykonané ako súčasť akreditovaného školenia alebo môžu byť vykonané nezávisle (napr. v skúšobnom centre, alebo v rámci verejnej skúšky). Absolvovanie akreditovaného školenia nie je nutným predpokladom pre skúšku.

Akreditácia

Lokálny výbor ISTQB môže akreditovať poskytovateľov školení, ktorých školiace materiály sú v súlade s touto učebnou osnovou. Poskytovatelia školení by mali obdržať akreditačné smernice od výboru, alebo osoby vykonávajúcej akreditáciu.

Akreditované školenie je uznané ako zodpovedajúce týmto osnovám a môže obsahovať ISTQB skúšku ako časť školenia.

Ďalšie rady pre poskytovateľov školení sa nachádzajú v Prílohe D.

Úroveň detailu

Úroveň detailu v tejto učebnej osnove umožňuje medzinárodne konzistentné vyučovanie a skúšanie. Aby bolo možné dosiahnuť tento cieľ, učebná osnova obsahuje:

- Všeobecné inštruktážne ciele popisujúce zámery základného stupňa.
- Zoznam informácií pre vyučovanie, vrátane popisu a odkazov na ďalšie zdroje ak sú požadované.
- Študijné ciele pre každú znalostnú oblasť, popisujúce požadovaný výsledok učenia sa a spôsob zmýšľania, ktorý je potrebné dosiahnuť.
- Zoznam výrazov, ktoré si študent musí zapamätať, a ktorým musí rozumieť.
- Popis kľúčových konceptov k vyučovaniu, vrátane zdrojov ako akceptovaná literatúra alebo štandardy.

Obsah tejto učebnej osnovy nie je popis celej znalostnej oblasti testovania softvéru. Odráža úroveň detailu, ktorý má byť pokrytý v školeniach pre základný stupeň.

Ako je táto učebná osnova usporiadaná

Osnovy majú šesť hlavných kapitol. Najvyššia úroveň nadpisu každej kapitoly ukazuje najvyššiu úroveň študijných cieľov, ktorá je v rámci kapitoly pokrytá a špecifikuje čas potrebný na kapitolu. Napríklad:

2. Testovanie v životnom cykle softvéru (Z2)	115 minút
--	-----------

Tento nadpis ukazuje, že kapitola 2 má študijné ciele Z1 a Z2 (Z1 sa vyžaduje pretože je zobrazený vyšší stupeň, Z3 sa nevyžaduje) a má určené trvanie 115 minút na zaškolenie kapitoly. Vo vnútri každej kapitoly sú rôzne počty sekcií. Každá sekcia má tiež študijný cieľ a požadovaný čas. Podsekcie, ktoré nemajú určený čas, sú zahrnuté v rámci času pre sekciu.

1. Základy testovania (Z2)

155 minút

Študijné ciele pre základy testovania

Tieto ciele určujú, čo budete vedieť po dokončení každého modulu.

1.1 Prečo je potrebné testovanie? (Z2)

ŠC-1.1.1 Popísať, spolu s príkladmi, spôsoby, akými môže defekt v softvéri spôsobiť škodu osobe, prostrediu alebo firme. (Z2)

ŠC-1.1.2 Rozlišovať medzi prvotnou príčinou defektu a jeho následkami. (Z2)

ŠC-1.1.3 Pomocou príkladov zdôvodniť, prečo je testovanie potrebné. (Z2)

ŠC-1.1.4 Popísať, prečo je testovanie súčasťou zabezpečenia kvality a uviesť príklady, ako testovanie prispieva k vyššej kvalite. (Z2)

ŠC-1.1.5 Pomocou príkladov vysvetliť a porovnať pojmy omyl, defekt, chyba, zlyhanie a súvisiaci pojem bug (Z2)

1.2 Čo je testovanie? (Z2)

ŠC-1.2.1 Zapamätať si všeobecné ciele testovania. (Z1)

ŠC-1.2.2 Uviesť príklady cieľov testovania v rôznych fázach životného cyklu softvéru. (Z2)

ŠC-1.2.3 Rozlišovať testovanie od ladenia (debugging) (Z2)

1.3 Sedem princípov testovania (Z2)

ŠC-1.3.1 Vysvetliť sedem princípov v testovaní. (Z2)

1.4 Základný testovací proces (Z1)

ŠC-1.4.1 Zapamätať si päť základných testovacích aktivít a príslušných úloh od plánovania až po uzavretie. (Z1)

1.5 Psychológia testovania (Z2)

ŠC-1.5.1 Zapamätať si psychologické faktory, ktoré ovplyvňujú úspech testovania (Z1)

ŠC-1.5.2 Posúdiť rozdielne postoje testera a vývojára. (Z2)

1.1 *Prečo je potrebné testovanie (Z2)*

20 minút

Základné výrazy

Bug, defekt, omyl, zlyhanie, chyba, kvalita, riziko.

1.1.1 Súvislosti v softvérových systémoch (Z1)

Softvérové systémy sú neoddeliteľnou súčasťou života, od obchodných aplikácií (napr. bankovníctvo) až po spotrebiteľské produkty (napr. autá). Väčšina ľudí už má skúsenosti so softvérom, ktorý nepracoval ako bolo očakávané. Softvér, ktorý nepracuje správne, môže spôsobiť veľa problémov, vrátane straty peňazí, času alebo obchodnej reputácie. Môže dokonca spôsobiť zranenie alebo smrť.

1.1.2 Príčiny softvérových defektov (Z2)

Človek sa môže dopustiť omylu, ktorý vyvolá defekt (chybu, bug) v programovom kóde alebo v dokumente. Ak sa vykoná kód obsahujúci defekt, systém nemusí urobiť čo by mal (resp. urobí niečo, čo by nemal) a tým spôsobí zlyhanie. Defekty v softvéri, systémoch alebo v dokumentoch môžu spôsobiť zlyhania, ale toto neplatí pri všetkých defektoch.

Defekty nastávajú, pretože ľudia sú omylní a pretože pracujú pod časovým tlakom, s komplexným kódom, v komplikovanej infraštruktúre, s meniacimi sa technológiami, a/alebo k nim prichádza kvôli vzájomnému pôsobeniu systémov.

Zlyhania môžu byť zapríčinené aj prírodnými podmienkami prostredia. Napríklad radiácia, magnetizmus, elektrické polia a znečistenie môže spôsobiť chyby vo firmvéri alebo ovplyvniť vykonávanie softvéru zmenou hardvérových podmienok.

1.1.3 Úloha testovania v softvérovom vývoji, údržbe a prevádzke (Z2)

Dôsledné testovanie softvéru a dokumentácie môže pomôcť znížiť riziko problémov, ktoré nastanú počas prevádzky a prispieva ku kvalite softvérového systému, ak sú zistené defekty opravené skôr, ako je systém uvoľnený do prevádzky.

Testovanie softvéru môže byť vyžadované tiež z dôvodu splnenia zmluvných alebo právnych požiadaviek, alebo priemyselných špecifických štandardov.

1.1.4 Testovanie a kvalita (Z2)

Pomocou testovania je možné merať kvalitu softvéru v zmysle zistených defektov, pre funkcionálne aj pre nefunkcionálne softvérové požiadavky a charakteristiky (napr. spoľahlivosť, použiteľnosť, účinnosť, udržateľnosť a prenositeľnosť). Pre ďalšie informácie o nefunkcionálnom testovaní pozri kapitolu 2; pre ďalšie informácie o softvérových charakteristikách pozri "Softvérové inžinierstvo – Kvalita softvérového produktu" (ISO 9126).

Testovanie môže dodať dôveru v kvalitu softvéru, ak nájde malé množstvo alebo žiadne defekty. Vhodne navrhnutý test, ktorý úspešne prejde, znižuje celkovú úroveň rizika v systéme. Keď testovanie preukáže defekty, kvalita softvérového systému sa zvýši, ak sú tieto defekty opravené.

Z predchádzajúcich projektov by sa mali brať ponaučenia. Pri pochopení prvotných príčin defektov zistených pri testovaní v iných projektoch môžu byť zlepšené procesy, ktoré by následne mali zabrániť ich zopakovaniu. Následkom toho sa zlepší kvalita budúcich systémov. Toto je aspekt zabezpečenia kvality.

Testovanie by malo byť integrované ako jedna z aktivít zabezpečenia kvality (t.j. popri vývojárskych štandardoch, školeniach a analýzach defektov).

1.1.5 Kedy je testovanie dostatočné? (Z2)

Rozhodnutie, kedy je testovanie dostatočné, by malo zohľadniť úroveň rizika, vrátane technických, bezpečnostných, obchodných rizík a projektových obmedzení, ako sú čas a rozpočet. Riziko je diskutované podrobnejšie v kapitole 5.

Testovanie by malo poskytnúť dostatok informácií kľúčovým osobám pre kvalifikované rozhodnutie o uvoľnení testovaného softvéru alebo systému do nasledujúceho stupňa vývoja alebo o odovzdaní zákazníkovi.

1.2 Čo je testovanie? (Z2)

30 minút

Základné výrazy

Ladenie, požiadavka, revízia, testovací prípad, testovanie, cieľ testovania.

Pozadie

Zvyčajne je testovanie vnímané len ako spúšťanie testov, t.j., že predstavuje spustenie softvéru. Toto je súčasť testovania, ale nie je to všetko z testovacích aktivít.

Testovacie aktivity existujú pred a po vykonaní testov. Tieto aktivity zahŕňajú plánovanie a riadenie, výber testovacích podmienok, navrhovanie a vykonanie testovacích prípadov, kontrola výsledkov, vyhodnocovanie výstupných kritérií, reportovanie testovacieho procesu a testovaného systému a finalizácia ukončovacích aktivít po ukončení fázy testovania. Testovanie tiež zahŕňa revidovanie dokumentov (vrátane zdrojového kódu) a vykonávanie statickej analýzy.

Statické aj dynamické testovanie môžu byť použité ako spôsoby pre dosiahnutie podobných cieľov. Poskytujú informácie, ktoré môžu byť použité na zlepšenie testovaného systému a procesov vývoja a testovania.

Testovanie môže mať nasledujúce ciele:

- nájdenie defektov;
- získanie dôvery s ohľadom na úroveň kvality
- poskytnutie informácií pre rozhodovanie;
- predchádzanie defektom.

Myšlienkový proces a činnosti spojené s návrhom testov dostatočne skoro v životnom cykle (verifikácia základu testovania prostredníctvom návrhu testov) môžu pomôcť zabrániť zaneseniu defektov do kódu. Revidovanie dokumentov (napr. požiadaviek), identifikácia a riešenie problémov taktiež pomáha pri predchádzaní defektom, ktoré by sa objavili v kóde.

Rôzne pohľady v testovaní zohľadňujú rôzne ciele. Napríklad vo vývojárskom testovaní (napr. komponentnom, integračnom a systémovom testovaní), môže byť hlavným cieľom spôsobenie toľkých zlyhaní, ako je len možné, za účelom identifikácie a opravy defektov v softvéri. V akceptačnom testovaní môže byť hlavným cieľom potvrdenie, že systém pracuje podľa očakávaní, aby sme tak získali dôveru, že je v súlade s požiadavkami. V niektorých prípadoch môže byť hlavným cieľom testovania ohodnotenie kvality softvéru (bez snahy opravovať defekty), dať informáciu kľúčovým osobám o rizikách uvoľnenia systému v danom čase. Testovanie údržby často zahŕňa testovanie s cieľom preveriť, či sa nezanesli ďalšie defekty počas vývoja zmien. Počas prevádzkového testovania môže byť hlavným cieľom zhodnotenie charakteristík systému, ako je spoľahlivosť alebo dostupnosť.

Je rozdiel medzi ladením a testovaním. Dynamické testovanie môže ukázať zlyhanie, ktoré sú spôsobené defektmi. Ladenie je vývojová aktivita, ktorá nachádza, analyzuje a odstraňuje príčinu zlyhaní. Následné retestovanie testerom slúži na uistenie sa, že oprava skutočne rieši zlyhanie. Zodpovednosť za tieto činnosti je zvyčajne taká, že tester testujú a vývojári ladia.

Proces testovania a testovacích činností sú popísané v časti 1.4.

1.3 Sedem princípov testovania (Z2)

35 minút

Základné výrazy

Vyčerpávajúce testovanie.

Princípy

Za posledných 40 rokov bolo navrhnutých niekoľko princípov testovania, ktoré poskytujú všeobecné pokyny spoločné pre každé testovanie.

Princíp 1 – Testovanie ukazuje prítomnosť defektov

Testovanie môže ukázať, že sú defekty prítomné, ale nemôže dokázať, že v softvéri nie sú žiadne defekty. Testovanie znižuje pravdepodobnosť, že v softvéri zostanú neobjavené defekty, avšak nenájdenie žiadneho defektu stále nie je dôkaz jeho bezchybnosti.

Princíp 2 – Vyčerpávajúce testovanie je nemožné

Testovanie všetkého (všetkých kombinácií vstupov a vstupných podmienok) nie je realizovateľné s výnimkou triviálnych prípadov. Namiesto vyčerpávajúceho testovania by mala byť k určeniu hlavného predmetu testovacieho úsilia použitá analýza rizík a stanovenie priorit.

Princíp 3 – Včasné testovanie

Pre včasnú identifikáciu defektov, musia testovacie aktivity začať v rámci životného cyklu vývoja softvéru alebo systému tak skoro, ako je to len možné a musia byť zamerané na definované ciele.

Princíp 4 – Zhlukovanie defektov

Testovacie úsilie musí byť zamerané proporčne na očakávanú a neskôr zistenú hustotu defektov v moduloch. Malé množstvo modulov zvyčajne obsahuje väčšinu defektov zistených počas testovania pred uvoľnením, alebo je zodpovedné za najviac prevádzkových zlyhaní.

Princíp 5 – Pesticídny paradox

Ak sú tie isté testy neustále opakované, časom ten istý súbor testovacích prípadov nenájde žiadne nové defekty. Na prekonanie tohto „pesticídového paradoxu“ je potrebné existujúce testovacie prípady pravidelne revidovať a upravovať a je potrebné napísať nové, odlišné testy na vykonanie iných častí softvéru alebo systému pre prípadné odhalenie ďalších defektov.

Princíp 6 – Testovanie je závislé na kontexte

Testovanie je vykonávané odlišne v rôznych kontextoch. Napríklad softvér kritický z pohľadu bezpečnosti sa testuje iným spôsobom ako webové stránky elektronického obchodu.

Princíp 7 – Falošná predstava o neexistencii omylov

Nájdenie a opravenie defektov nepomôže, ak je vytvorený systém nepoužiteľný a nespĺňa potreby a očakávania užívateľov.

1.4 Základný testovací proces (Z1)

35 minút

Základné výrazy

Konfirmačné testovanie, retestovanie, výstupné kritérium, incident, regresné testovanie, základ testovania, testovacia podmienka, pokrytie testovania, testovacie dáta, vykonanie testu, protokol o teste, testovací plán, testovacia procedúra, pravidlá testovania, zostava testovania, sumárna správa testovania, testvér.

Pozadie

Najviac viditeľnou časťou testovania je vykonávanie testov. Ale aby bolo efektívne a účinné, mali by testovacie plány taktiež obsahovať čas na plánovanie testov, navrhovanie testovacích prípadov, prípravu pre vykonávanie testov a vyhodnocovanie výsledkov.

Základný testovací proces pozostáva z nasledujúcich hlavných aktivít:

- Plánovanie a riadenie testovania;
- Analýza a návrh testovania;
- Implementácia a vykonávanie testovania;
- Vyhodnotenie výstupných kritérií a reportovanie;
- Činnosti súvisiace s ukončením testovania.

Hoci sú vymenované aktivity logicky postupné, môžu sa v testovacom procese prekrývať, alebo sa môžu vykonávať súbežne. Zvyčajne je vyžadované prispôbenie týchto hlavných aktivít testovania v rámci kontextu systému a projektu.

1.4.1 Plánovanie a riadenie testovania (Z1)

Plánovanie testovania je aktivita definovania cieľov testovania a špecifikácie testovacích činností, za účelom dosiahnutia cieľov a poslania.

Riadenie testovania je priebežná aktivita porovnávajúca aktuálny postup oproti plánu a reportovanie stavu, vrátane odchýliek od plánu. Zahŕňa realizáciu činností nevyhnutných na dosiahnutie poslania a cieľov projektu. Aby mohlo byť testovanie riadené, mali by byť činnosti testovania sledované v priebehu projektu. Plánovanie testovania berie do úvahy aj spätnú väzbu z monitorovacích a riadiacich aktivít.

Plánovanie testovania a riadiace úlohy sú definované v kapitole 5 tejto učebnej osnovy.

1.4.2 Analýza a návrh testovania (Z1)

Analýza a návrh testovania je činnosť, počas ktorej sú všeobecné ciele testovania transformované do konkrétnych testovacích podmienok a testovacích prípadov.

Hlavné činnosti analýzy a návrhu testovania sú:

- Revidovanie základov testovania (ako sú požiadavky, úroveň integrity softvéru¹ (úroveň rizika), reporty analýzy rizík, architektúra, návrh, špecifikácie rozhraní).
- Vyhodnotenie testovateľnosti základu testovania a objektov testovania.

¹ Stupeň, ktorý softvér spĺňa, alebo musí spĺňať s ohľadom na sadu softvéru zvolenú kľúčovými XXX a / alebo systémových softvérových charakteristik (napr. zložitosť softvéru, vyhodnotenie rizík, úroveň bezpečnosti, úroveň zabezpečenia, požadovaná výkonnosť, spoľahlivosť, alebo cena), ktoré sú definované, aby odrážali dôležitosť softvéru pre jeho XXX.

- Identifikovanie a stanovenie priorít testovacích podmienok, ktoré je založené na analýze testovacích položiek, špecifikácie, správaní a štruktúre softvéru.
- Navrhovanie a stanovenie priorít vysoko úrovňových (high-level) testovacích prípadov.
- Identifikácia potrebných testovacích dát podporujúcich testovacie podmienky a testovacie prípady.
- Navrhovanie nastavenia testovacieho prostredia a identifikovanie požadovanej infraštruktúry a nástrojov.
- Vytvorenie obojsmernej sledovateľnosti medzi základom testovania a testovacími prípadmi

1.4.3 Implementácia a vykonanie testovania (Z1)

Implementácia a vykonanie testovania je činnosť, v priebehu ktorej sú testovacie procedúry alebo skripty špecifikované kombinovaním testovacích prípadov v určitom poradí a zahŕňajú všetky ďalšie informácie potrebné pre vykonanie testu, je nastavené testovacie prostredie a testy sú spustené.

Hlavné činnosti implementácie a vykonania testov sú:

- Finalizácia, implementácia a stanovenie priorít testovacích prípadov (vrátane identifikácie testovacích dát).
- Vytvorenie a stanovenie priorít testovacích procedúr, vytvorenie testovacích dát a voliteľne príprava testovacích postrojov a písania automatizovaných testovacích skriptov.
- Vytvorenie zostáv testov z testovacích procedúr pre účinné vykonávanie testov.
- Overenie, či je testovacie prostredie nastavené korektne.
- Verifikácia a aktualizácia obojsmernej sledovateľnosti medzi základom testovania a testovacími prípadmi
- Vykonanie testovacích procedúr či už manuálne alebo s použitím nástrojov pre vykonanie testov, podľa plánovaného poradia.
- Zaznamenávanie výsledkov vykonaných testov a zaznamenávanie označenia a verzií testovaného softvéru, testovacích nástrojov a testvéru.
- Porovnávanie skutočných výsledkov s očakávanými výsledkami.
- Reportovanie nezhôd ako incidentov a analýza pre stanovenie ich príčiny (napr. defekt v kóde, vurčených testovacích dátach, v testovacom dokumente alebo chyba v spôsobe, akým bol test vykonaný).
- Opakovanie testovacích aktivít ako výsledok prijatia opatrenia pre každú nezrovnalosť. Napríklad opätovné vykonanie testu, ktorý predtým skončil s chybou, kvôli potvrdeniu opravy (konfirmačné testovanie), vykonanie opraveného testu a/alebo vykonanie testov s cieľom uistiť sa, že v nezmenených oblastiach softvéru neboli zanesené defekty, alebo že oprava defektu neodhalila ďalšie defekty (regresné testovanie).

1.4.4 Vyhodnotenie výstupných kritérií a reportovanie (Z1)

Vyhodnotenie výstupných kritérií je aktivita, kde je vykonanie testov hodnotené voči definovaným cieľom. Malo by byť vykonané pre každú úroveň testovania (viď kapitola 2.2).

Vyhodnotenie výstupných kritérií má nasledovné hlavné činnosti:

- Kontrola protokolov o testoch voči výstupným kritériám špecifikovaným v plánovaní testovania.
- Zhodnotenie, či sú potrebné ďalšie testy, alebo či by špecifikované výstupné kritériá mali byť zmenené.
- Napísanie súhrnnej správy o testovaní pre kľúčové osoby.

1.4.5 Aktivity uzatvorenia testu (Z1)

Aktivity uzatvorenia testu zhromažďujú dáta z ukončených testovacích aktivít za účelom konsolidácie skúseností, testvéru, faktov a čísel. K aktivitám uzatvorenia testu dochádza v rámci projektových

mílnikov, napríklad po uvoľnení softvérového systému, po ukončení testovacieho projektu (alebo jeho zrušení), po dosiahnutí mílnika alebo ak bol ukončený release prevádzkovej verzie.

Aktivity uzatvorenia testu zahŕňajú nasledovné hlavné činnosti:

- Kontrola, ktoré plánované dodávky boli dodané.
- Uzavretie záznamov o incidentoch alebo nahlásenie záznamov o zmene pre tie, ktoré zostali otvorené.
- Dokumentácia akceptácie systému.
- Finalizovanie a archivovanie testvéru, testovacieho prostredia a testovacej infraštruktúry pre neskoršie opätovné použitie.
- Odovzdanie testvéru organizácii zabezpečujúcej prevádzku.
- Analyzovanie získaných ponaučení na určenie zmien potrebných pre budúce releasy a projekty
- Použitie získaných informácií na zlepšenie zrelosti testovania.

1.5 *Psychológia testovania (Z2)*

25 minút

Základné výrazy

Odhadovanie omylov, nezávislosť.

Pozadie

Prístup, ktorý má byť použitý počas testovania a revidovania je odlišný od toho, ktorý sa používa pri vývoji softvéru. So správnym prístupom sú vývojári schopní testovať ich vlastný kód, ale oddelenie tejto zodpovednosti na testera sa zväčša robí s cieľom sústrediť úsilie a poskytnúť ďalšie benefity, ako je nezávislý pohľad trénovanými a profesionálnymi zdrojmi z testovania. Nezávislé testovanie môže byť realizované na ktorejkoľvek úrovni testovania.

Určitý stupeň nezávislosti (čím sa vyhneme zaujatosti autora) často umožní testerovi byť viac efektívnym pri hľadaní defektov a zlyhaní. Avšak nezávislosť nie je náhrada za znalosť (systému) - vývojári môžu účinne nájsť množstvo defektov vo svojom vlastnom kóde. Môže byť definovaných niekoľko stupňov nezávislosti ako je uvedené nižšie, od najnižšieho stupňa po najvyšší:

- Testy navrhnuté osobou (osobami), ktoré píšú testovaný softvér (nízky stupeň nezávislosti).
- Testy navrhnuté inou osobou (osobami) (napr. z vývojárskeho tímu).
- Testy navrhnuté osobou (osobami) z inej organizačnej skupiny (napr. nezávislý testovací tím), alebo špecialistami na testovanie (napr. špecialistami na testovanie použiteľnosti alebo výkonu).
- Testy navrhnuté osobou (osobami) z inej organizácie alebo spoločnosti (t.j. outsourcing alebo certifikácia externou inštitúciou).

Ľudia a projekty sú riadené cieľmi. Ľudia majú tendenciu zosúladiť svoje plány s cieľmi stanovenými manažmentom alebo inými kľúčovými osobami, napríklad nájsť defekty alebo potvrdiť, že softvér funguje. Preto je dôležité jasné stanovenie cieľov testovania.

Identifikovanie zlyhaní počas testovania môže byť vnímané ako kritika voči produktu a voči autorovi. Testovanie je z tohto dôvodu často vnímané ako deštruktívna aktivita, hoci je veľmi konštruktívna v manažmente produktových rizík. Hľadanie zlyhaní v aplikácii vyžaduje zvedavosť, profesionálny pesimizmus, kritický pohľad, zameranie na detaily, dobrú komunikáciu s partnermi-vývojárami a skúsenosť, na základe ktorej možno odhadovať omyly.

Ak sú omyly, defekty alebo zlyhania komunikované konštruktívnym spôsobom, je možné vyhnúť sa zlému pocitu medzi testerami a analytikmi, návrhármi a vývojárami. Toto platí pre defekty nájdené počas revízií, rovnako ako pre testovanie.

Tester a vedúci tester potrebujú dobré medziľudské zručnosti aby boli schopní komunikovať faktické informácie o defektoch, pokroku a rizikách, konštruktívnym spôsobom. Tvorcovi softvéru alebo dokumentu môže informácia o defekte pomôcť zlepšiť jeho zručnosti. Defekty zistené a opravené počas testovania ušetria neskôr čas a peniaze a znižujú riziká.

Komunikačné problémy môžu nastávať, hlavne ak sú tester vnímaní len ako hlásatelia nechcených správ o defektoch. Avšak existuje niekoľko spôsobov na zlepšenie komunikácie a vzťahov medzi testerami a ostatnými členmi:

- Začať so spoluprácou radšej ako bojovať – pripomenúť každému spoločný cieľ lepšej kvality systémov.
- Komunikovať zistenia o produkte nestranným, vecne orientovaným spôsobom, bez kritiky zodpovednej osoby, napríklad písať objektívne a faktické správy o incidentoch a zistenia z revidovania.
- Pokúsiť sa porozumieť tomu, ako sa cítia iné osoby a prečo je ich reakcia taká, aká je.
- Uistiť sa, že druhá osoba porozumela tomu, čo ste povedali, a naopak.

1.6 Etický kódex

10 minút

Zapojenie jednotlivcov do testovania softvéru im umožňuje dozvedieť sa dôverné a chránené informácie. Etický kódex je nevyhnutný, okrem iného, aj na zabezpečenie toho, že tieto informácie nie sú neprimerane použité. Uznávajúc etický kódex ACM a IEEE pre inžinierov, ISTQB® deklaruje nasledujúci etický kódex:

VEREJNOSŤ - Certifikovaní testerí softvéru musia konať v súlade s verejným záujmom.

KLIENT A ZAMESTNÁVATEĽ - Certifikovaní testerí softvéru musia konať spôsobom, ktorý je v najlepšom záujme svojho klienta a zamestnávateľa, v súlade s verejným záujmom.

PRODUKT - Certifikovaní testerí softvéru musia zabezpečiť, že výstupy, ktoré poskytujú (na produktoch a systémoch, ktoré testujú) spĺňajú najvyššie možné profesionálne štandardy.

SÚDNOSŤ - Certifikovaní testerí softvéru si musia zachovávať bezúhonnosť a nezávislosť pri svojom profesionálnom posudzovaní.

MANAŽMENT - Certifikovaní test manažéri a vedúci sa musia hlásiť k etickému prístupu k manažmentu testovania softvéru a propagovať ho.

PROFESIA - Certifikovaní testerí softvéru musia zvyšovať dôveryhodnosť a dobrú povest' svojej profesie v súlade s verejným záujmom.

KOLEGOVIA - Certifikovaní testerí softvéru musia byť ku svojim kolegom spravodliví, podporovať ich a napomáhať pri spolupráci s vývojármi softvéru.

OSOBNÝ ROZVOJ - Certifikovaní testerí softvéru sa musia podieľať na celoživotnom vzdelávaní v rámci vykonávania ich profesie a musia propagovať etický prístup k výkonu povolania.

Referencie

- 1.1.5 Black, 2001, Kaner, 2002
- 1.2 Beizer, 1990, Black, 2001, Myers, 1979
- 1.3 Beizer, 1990, Hetzel, 1988, Myers, 1979
- 1.4 Hetzel, 1988
- 1.4.5 Black, 2001, Craig, 2002
- 1.5 Black, 2001, Hetzel, 1988

2. Testovanie v životnom cykle softvéru (Z2)

115 minút

Študijné ciele pre testovanie naprieč životným cyklom softvéru

Tieto ciele určujú, čo budete vedieť po dokončení každého modulu.

2.1 Modely softvérového vývoja (Z2)

ŠC-2.1.1 Vysvetliť vzťah medzi vývojom, testovacími aktivitami a pracovnými produktmi v životnom cykle vývoja na použití typov projektov a produktov (Z2)

ŠC-2.1.2 Chápať fakt, že modely vývoja softvéru musia byť prispôbené kontextu projektu a produktovým charakteristikám. (Z1)

ŠC-2.1.3 Zapamätať si charakteristiky dobrého testovania, ktoré sú použiteľné v akomkoľvek modeli životného cyklu. (Z1)

2.2 Úrovne testovania (Z2)

ŠC-2.2.1 Porovnať rôzne úrovne testovania: hlavné ciele, typické objekty testovania, typické ciele testovania (napr. funkcionálne alebo štrukturálne) a súvisiace pracovné produkty, ľudia, ktorí testujú, typy defektov a zlyhaní, ktoré majú byť identifikované. (Z2)

2.3 Typy testov (Z2)

ŠC-2.3.1 Porovnať štyri typy testovania softvéru (funkcionálne, nefunkcionálne, štrukturálne, testovanie zmien) na základe príkladu. (Z2)

ŠC-2.3.2 Chápať, že funkcionálne a štrukturálne testy sa vyskytujú na ktorejkoľvek úrovni testovania. (Z1)

ŠC-2.3.3 Identifikovať a popísať nefunkcionálne typy testovania založené na nefunkcionálnych požiadavkách. (Z2)

ŠC-2.3.4 Identifikovať a popísať typy testovania založené na analýze štruktúry alebo architektúry softvérového systému. (Z2)

ŠC-2.3.5 Popísať účel konfirmačného testovania a regresného testovania. (Z2)

2.4 Testovanie údržby (Z2)

ŠC-2.4.1 Porovnať testovanie údržby (testovanie existujúceho systému) s testovaním novej aplikácie s ohľadom na typy testov, dôvody spustenia testu a objem testovania. (Z2)

ŠC-2.4.2 Rozoznať indikátory pre testovanie údržby (modifikácia, migrácia a vyradenie). (Z1)

ŠC-2.4.3 Popísať rolu regresného testovania a dopadovú analýzu v oblasti údržby. (Z2)

2.1 Modely softvérového vývoja (Z2)

20 minút

Základné výrazy

Krabicový softvér, iteratívno-inkrementálny vývojový model, validácia, verifikácia, V-model.

Pozadie

Testovanie neexistuje v izolácii; testovacie aktivity sa vzťahujú k aktivitám vývoja softvéru. Rozdielne modely životného cyklu vývoja vyžadujú rozdielne prístupy k testovaniu.

2.1.1 V-model (sekvenčný vývojový model) (Z2)

I keď existujú rôzne varianty V-modelu, bežný typ V-modelu používa štyri úrovne testovania korešpondujúce so štyrmi úrovňami vývoja.

Štyri úrovne používané v tejto učebnej osnove sú:

- testovanie komponentov (jednotiek);
- integračné testovanie;
- systémové testovanie;
- akceptačné testovanie.

V skutočnosti môže mať V-model viac alebo menej úrovní resp. rozdielne úrovne vývoja a testovania v závislosti na projekte a softvérovom produkte. Napríklad po testovaní komponentov môže nasledovať integračné testovanie komponentov a po systémovom testovaní systémové integračné testovanie.

Softvérové pracovné produkty vytvorené počas vývoja (ako napríklad biznis scenáre alebo prípady použitia, špecifikácie požiadaviek, dokumenty návrhu a kód) sú často základom pre testovanie na jednej alebo viacerých úrovniach testovania. Referencie na všeobecné pracovné produkty zahŕňa Capability Maturity Model Integration (CMMI) alebo 'Procesy životného cyklu vývoja softvéru' (IEEE/IEC 12207). Verifikácia a validácia (a skorý návrh testov) môžu byť vykonané počas vývoja softvérových pracovných produktov.

2.1.2 Iteratívno-inkrementálne vývojové modely (Z2)

Iteratívno-inkrementálny vývoj je proces zavedenia požiadaviek, navrhovania, tvorby a testovania systému v sérii krátkych vývojových cyklov. Príkladmi sú: prototypovanie, RAD (Rapid Application Development), RUP (Rational Unified Process) a agilné vývojové modely. Systém ktorý je vytvorený použitím týchto modelov možno testovať na viacerých úrovniach testovania počas každej iterácie. Inkrement pridaný k iným, skôr vyvinutým, formuje rastúci neúplný systém, ktorý by mal byť takisto testovaný. Regresné testovanie je stále viac dôležitejšie vo všetkých iteráciách nasledujúcich po prvej. Verifikácia a validácia môžu byť vykonané pre každý inkrement.

2.1.3 Testovanie v modeli životného cyklu (Z2)

V akomkoľvek modeli životného cyklu je niekoľko charakteristík dobrého testovania:

- Pre každú vývojovú aktivitu existuje zodpovedajúca testovacia aktivita.
- Každá úroveň má ciele testovania špecifické pre túto úroveň.
- Analýza a návrh testov pre danú úroveň testovania by mali začať počas zodpovedajúcej vývojovej aktivity.
- Tester by mali byť zahrnutí do revidovania dokumentov hneď v okamihu, keď sú k dispozícii pracovné verzie v životnom cykle vývoja.

Certifikovaný tester

Učebná osnova pre základný stupeň



Úrovne testovania môžu byť kombinované alebo reorganizované v závislosti na charaktere projektu alebo architektúre systému. Napríklad pri integrácii krabicového softvéru do systému môže kupujúci vykonať integračné testovanie na systémovej úrovni (napr. integrácia do infraštruktúry a iných systémov alebo nasadenie systému) a akceptačné testovanie (funkcionálne a/alebo nefunkcionálne a užívateľské a/alebo prevádzkové testovanie).

2.2 Úroveň testovania (Z2)

40 minút

Základné výrazy

Alfa testovanie, beta testovanie, testovanie komponentu, ovládač, testovanie "v teréne", funkcionálna požiadavka, integrácia, integračné testovanie, nefunkcionálna požiadavka, testovanie robustnosti, nástavec systémové testovanie, testovacie prostredie, úroveň testovania, vývoj riadený testovaním, užívateľské akceptačné testovanie.

Pozadie

Pre každú z úrovní testovania môže byť identifikované nasledovné všeobecné ciele, pracovný(é) produkt(y), na ktorý(é) sa odkazuje pri odvídzaní testovacích prípadov (t.j. základ testovania), objekt testovania (t.j. čo je testované), typické defekty a zlyhania, ktoré môžu byť nájdené, požiadavky na testovací postroj a podporu nástrojov, špecifické prístupy a zodpovednosti.

Testovanie konfiguračných dát systému musí byť zvažované počas plánovania testov.

2.2.1 Testovanie komponentov (Z2)

Základ testovania:

- Požiadavky na komponenty.
- Detailný návrh.
- Kód.

Typické objekty testovania:

- Komponenty.
- Programy.
- Konverzia dát / migračné programy.
- Databázové moduly.

Testovanie komponentov (tiež známe ako jednotkové testovanie, testovanie modulu alebo testovanie programu), hľadá defekty vnútri softvérových komponentov a verifikuje fungovanie softvérových komponentov, modulov, programov, objektov, tried, atď., ktoré sú testovateľné samostatne. Môže byť vykonané v izolácii od zvyšku systému v závislosti na kontexte životného cyklu vývoja a systému. Pri testovaní komponentov môžu byť použité nástavce, ovládače a simulátory.

Testovanie komponentov môže zahŕňať testovanie funkcionality a špecifických nefunkcionálnych charakteristík ako sú správanie zdrojov (napr. hľadanie pretečení pamäte) alebo testovanie robustnosti ako aj štruktúrne testovanie (napr. pokrytie rozhodovaní). Testovacie prípady sú odvodené z pracovných produktov ako špecifikácia komponentu, návrh softvéru alebo dátový model.

Zvyčajne sa testovanie komponentov realizuje s prístupom ku kódu, ktorý je testovaný a s podporou vývojového prostredia ako napríklad framework jednotkového testovania alebo nástroj pre ladenie. V praxi je obvykle testovanie komponentov vykonávané za účasti programátora, ktorý kód napísal. Defekty sú obvyčajne opravované hneď, ako sú nájdené, bez formálneho riadenia týchto defektov.

Jeden z prístupov v testovaní komponentov je príprava a automatizácia testovacích prípadov pred kódovaním. Nazýva sa "testovanie na prvom mieste" alebo vývoj riadený testovaním. Tento prístup je vysoko iteratívny a je založený na cykloch, v ktorých sa vyvíjajú testovacie prípady, následne sa vytvárajú a integrujú malé časti kódu, vykonávajú sa testy komponentov a opravujú sa všetky problémy. Toto iteruje až do doby, pokiaľ testy nie sú úspešné.

2.2.2 Integrované testovanie (Z2)

Základ testovania:

- Softvérový a systémový návrh.
- Architektúra.
- Pracovné toky (workflows).
- Prípady použitia.

Typické objekty testovania:

- Podsystemy.
- Implementácie databáz.
- Infraštruktúra.
- Rozhrania.
- Konfigurácia systému a konfiguračné dáta.

Integrované testovanie testuje rozhrania medzi komponentmi, interakcie s rôznymi časťami systému ako sú operačný systém, súborový systém, hardvér a rozhrania medzi systémami.

Môže existovať viac ako jedna úroveň integrovaného testovania a integrované testovanie môže byť vykonané na objektoch testovania rôznej veľkosti nasledovne:

1. Integrované testovanie komponentov preveruje interakciu medzi komponentmi softvéru a je vykonávané po testovaní komponentov;
2. Systémové integrované testovanie preveruje interakciu medzi rôznymi systémami alebo medzi hardvérom a softvérom a môže byť vykonané po systémovom testovaní. V tomto prípade môže vyvíjajúca organizácia kontrolovať iba jednu stranu rozhrania. Toto môže byť posúdené ako riziko. Biznis procesy implementované ako workflow môžu zahŕňať série systémov. Problémy na úrovni viacerých platforiem môžu byť závažné.

S väčším záberom integrácie je stále zložitejšie izolovať defekty k špecifickému komponentu alebo systému, čo môže viesť k zvýšeniu rizika a dodatočného času na riešenie problémov.

Systematické integrované stratégie môžu byť založené na architektúre systému (ako zhora-nadol, zdola-nahor), funkcionálnych úlohách, sekvenciách spracovania transakcií alebo na iných aspektoch systému alebo komponentov. Aby bolo jednoduchšie izolovať chyby a skoro odhaliť defekty, integrácia by mala byť realizovaná ako inkrementálna namiesto "veľkým treskom".

V integrovanom testovaní môže byť zahrnuté testovanie špecifických nefunkcionálnych charakteristík (napr. výkon) ako aj funkcionálne testovanie.

V každom štádiu integrácie sa tester zameriavajú výhradne na samotnú integráciu. Napríklad ak integrujú modul A s modulom B, zaujímajú sa o testovanie komunikácie medzi modulmi, nie o funkcionálnosť samotného modulu, keďže táto bola testovaná v rámci testovania komponentov. Môžu byť použité oba prístupy: funkcionálny aj štruktúrny.

V ideálnom prípade by tester mali rozumieť architektúre a ovplyvňovať integrované plánovanie. Ak sú integrované testy plánované pred tým, ako sú vytvorené komponenty alebo systémy, tieto komponenty môžu byť vytvorené v takom poradí, aby bolo testovanie čo najúčinnejšie.

2.2.3 Systémové testovanie (Z2)

Základ testovania:

- Špecifikácia požiadaviek systému a softvéru.
- Prípady použitia.
- Funkcionálna špecifikácia.

- Reporty analýzy rizík.

Typické objekty testovania:

- Systémové, používateľské a prevádzkové manuály.
- Konfigurácia systému a konfiguračné dáta.

Systémové testovanie sa zaoberá správaním celého systému/produktu. Rozsah testu musí byť jasne vyznačený v hlavnom a úrovňovom testovacím pláne pre danú úroveň testovania.

V systémovom testovaní by malo testovacie prostredie korešpondovať s finálnym cieľovým alebo produkčným prostredím v čo najväčšej možnej miere za účelom minimalizácie rizika, že zlyhania, ktoré sú špecifické pre prostredie nebudú pri testovaní nájdené.

Systémové testovanie môže zahŕňať testy založené na rizikách a/alebo na špecifikáciách požiadaviek, biznis procesoch, prípadoch použitia alebo iných vysoko-úrovňových slovných popisoch alebo modeloch správania systému, interakciách s operačným systémom a na systémových zdrojoch.

Systémové testovanie by malo preskúmať funkcionálne aj nefunkcionálne požiadavky systému a kvalitatívne charakteristiky dát. Tester sa tiež musia vysporiadať s nekompletnými alebo nedokumentovanými požiadavkami. Systémové testovanie funkcionálnych požiadaviek začína používaním najvhodnejších techník založených na špecifikácii (čierna skrinka) vzhľadom na aspekt systému, ktorý bude testovaný. Napríklad môže byť vytvorená rozhodovacia tabuľka pre kombinácie následkov popísaných v biznis pravidlách. Techniky založené na štruktúre (biela skrinka) môžu byť potom použité na zhodnotenie dôkladnosti testovania s ohľadom na štrukturálny element, ako napríklad štruktúra menu alebo navigácia web stránky. (Vid' kapitola 4.)

Systémové testovanie často realizuje nezávislý testovací tím.

2.2.4 Akceptačné testovanie (Z2)

Základ testovania:

- Používateľské požiadavky.
- Systémové požiadavky.
- Prípady použitia.
- Biznis procesy.
- Reporty analýzy rizík.

Typické objekty testovania:

- Biznis procesy na plne integrovanom systéme.
- Prevádzkové a údržbové procesy.
- Používateľské postupy.
- Formuláre.
- Reporty.
- Konfiguračné dáta.

Akceptačné testovanie je často zodpovednosťou zákazníkov alebo užívateľov systému; rovnako do neho môžu byť zapojené aj ďalšie kľúčové osoby.

V akceptačnom testovaní je cieľom nastolenie dôvery k systému, jeho častiam alebo špecifickým nefunkcionálnym charakteristikám systému. V akceptačnom testovaní nie je hlavným účelom nájdenie defektov. Akceptačné testovanie môže ohodnotiť pripravenosť systému pre nasadenie a používanie, i keď nie je nevyhnutne poslednou úrovňou testovania. Napríklad integračné testovanie rozsiahlych systémov môže byť vykonané po akceptačnom testovaní systému.

Akceptačné testovanie sa môže vyskytnúť vo viacerých časoch v životnom cykle, napríklad:

- Krabicový softvérový produkt môže byť akceptačne testovaný, keď je inštalovaný alebo integrovaný.
- Akceptačné testovanie použiteľnosti komponentu môže byť vykonávané počas testovania komponentu.
- Akceptačné testovanie nového funkcionálneho zlepšenia môže byť realizované pred systémovým testovaním.

Typické formy akceptačného testovania sú nasledujúce:

Užívateľské akceptačné testovanie

Zvyčajne verifikuje pripravenosť na použitie systému biznis užívateľmi.

Prevádzkové (akceptačné) testovanie

Akceptácia systému systémovými administrátormi, zahŕňa:

- testovanie zálohy/obnovy;
- regenerácia po havárii;
- správu užívateľov;
- úlohy údržby;
- načítavanie dát a migračné úlohy
- pravidelnú kontrolu bezpečnostných nedostatkov.

Zmluvné a regulačné akceptačné testovanie

Zmluvné akceptačné testovanie je vykonávané voči zmluvným akceptačným kritériám pre produkciu softvéru vyvinutého na zákazku. Akceptačné kritériá by mali byť definované v dobe, keď zúčastnené strany odsúhlasia kontrakt.

Regulačné testovanie je vykonávané voči akýmkoľvek predpisom, ktoré musia byť dodržané, ako napríklad vládne, právne alebo bezpečnostné predpisy.

Alfa testovanie a beta testovanie (alebo testovanie "v teréne")

Vývojári trhového alebo krabicového softvéru chcú často dostať spätnú väzbu od potenciálnych alebo existujúcich zákazníkov na ich trhu pred tým, ako je softvér daný do komerčného predaja. Alfa testovanie je vykonávané na pôde vyvíjajúcej organizácie, ale nie vývojovým tímom. Beta testovanie alebo testovanie "v teréne" je vykonávané zákazníkmi alebo potenciálnymi zákazníkmi v ich vlastnom prostredí.

Organizácie môžu používať aj iné termíny, ako napríklad „podnikové akceptačné testovanie“ a „akceptačné testovanie u zákazníka“ pre systémy testované pred tým a po tom, ako sú prenesené na stranu zákazníka.

2.3 Typy testov (Z2)

40 minút

Základné výrazy

Testovanie čiernej skrinky, pokrytie kódu, funkcionálne testovanie, testovanie spolupôsobenia (interoperability), záťažové testovanie, testovanie udržateľnosti, testovanie výkonu, testovanie prenositeľnosti, testovanie spoľahlivosti, testovanie bezpečnosti, stres testovanie, štruktúrne testovanie, testovanie použiteľnosti, testovanie bielej skrinky.

Pozadie

Skupina testovacích aktivít môže byť zameraná na verifikáciu softvérového systému (alebo jeho časti) na základe špecifického dôvodu alebo cieľa testovania.

Každý typ testu je zameraný na určitý účel testovania, ktorým môže byť niektorý z týchto:

- funkcia, ktorá má byť vykonávaná softvérom;
- nefunkcionálna kvalitatívna charakteristika ako napríklad spoľahlivosť či použiteľnosť;
- štruktúra alebo architektúra softvéru alebo systému;
- súvisiaca zmena, t.j. potvrdenie, že defekty boli opravené (konfirmačné testovanie) a hľadanie neúmyselných zmien (regresné testovanie).

Model softvéru môže byť vyvinutý a/alebo použitý v štruktúrnom testovaní (napr. model riadiaceho toku alebo model štruktúry menu), nefunkcionálnom testovaní (napr. model výkonnosti, model použiteľnosti, modelovanie bezpečnostných hrozieb) a funkcionálnom testovaní (napr. model procesných tokov, model prechodu stavov alebo jednoduchá slovná špecifikácia).

2.3.1 Testovanie funkcionality (funkcionálne testovanie) (Z2)

Funkcie, ktoré má systém, subsystém alebo komponent vykonávať, môžu byť popísané v pracovných produktoch, ako sú špecifikácie požiadaviek, prípady použitia alebo funkcionálna špecifikácia alebo môžu byť nedokumentované. Funkcie predstavujú "čo" systém robí.

Funkcionálne testy sú založené na funkciách a vlastnostiach (ako sú popísané v dokumentoch alebo chápané testerom) a ich spolupôsobení so špecifickými systémami. Môžu byť vykonávané na všetkých úrovniach testovania (napr. testy komponentov môžu byť založené na špecifikácii komponentov).

Techniky založené na špecifikácii môžu byť použité s cieľom odvodit' testovacie podmienky a testovacie prípady z funkcionality softvéru alebo systému (viď kapitola 4). Funkcionálne testovanie skúma externé správanie softvéru (testovanie čiernej skrinky).

Jeden z typov funkcionálneho testovania, testovanie bezpečnosti, skúma funkcie (napr. firewall) vzťahujúce sa k detekcii hrozieb ako napríklad vírusy zlomyseľných outsiderov. Ďalší typ funkcionálneho testovania – testovanie spolupôsobenia (interoperability) – hodnotí schopnosť interakcie softvérového produktu s jedným alebo viacerými určenými komponentmi alebo systémami.

2.3.2 Testovanie nefunkcionálnych softvérových charakteristík (nefunkcionálne testovanie) (Z2)

Nefunkcionálne testovanie zahŕňa testovanie výkonu, záťažové testovanie, stres testovanie, testovanie použiteľnosti, testovanie udržateľnosti, testovanie spoľahlivosti a testovanie prenositeľnosti, ale neobmedzuje sa len na ne. Je testovaním toho "ako" systém pracuje.

Nefunkcionálne testovanie môže byť vykonávané na všetkých úrovniach testovania. Termín nefunkcionálne testovanie popisuje testy požadované na meranie charakteristík systému a softvéru, ktoré môžu byť kvantifikované voči rôznym stupniciam merania, ako napríklad doby odozvy pre testovanie výkonu. Tieto testy sa môžu odvolávať na kvalitatívne modely ako napríklad na model definovaný v štandarde 'Softvérové inžinierstvo – Kvalita softvérového produktu' (ISO 9126). Ne-funkcionálne testovanie zvažuje externé správanie softvéru a vo väčšine prípadov k tomu používa techniky návrhu testovania čiernej skrinky.

2.3.3 Testovanie štruktúry/architektúry softvéru (štrukturálne testovanie) (Z2)

Štrukturálne testovanie (biela skrinka) môže byť vykonávané na všetkých úrovniach testovania. Štrukturálne techniky sú najlepšie použité po uplatnení techník založených na špecifikácii, za účelom pomôcť merať dôkladnosť testovania pomocou stanovenia pokrytia typu štruktúry.

Pokrytie je miera, do akej je štruktúra preverená testovacou sadou. Vyjadrená je ako percento položiek, ktoré daná sada pokrýva. Ak nie je pokrytie 100%, potom môžu byť pre tie položky, ktoré chýbajú, navrhnuté ďalšie testy, ktoré ich otestujú za účelom zvýšenia pokrytia. Techniky pokrytia sú obsahom kapitoly 4.

Na všetkých úrovniach testovania, avšak obzvlášť pri testovaní komponentov a integračnom testovaní komponentov, môžu byť použité nástroje na meranie pokrytia elementov kódu, ako napríklad príkazov alebo rozhodovaní. Štrukturálne testovanie môže byť založené na architektúre systému, ako napríklad hierarchia volaní častí systému.

Prístupy štrukturálneho testovania môžu byť aplikované aj na systémovej, systémovej integračnej alebo akceptačnej úrovni testovania (napr. na biznis modely alebo štruktúry menu).

2.3.4 Testovanie súvisiace so zmenami: retestovanie a regresné testovanie (Z2)

Po tom, ako je defekt nájdený a opravený, by mal byť softvér retestovaný za účelom potvrdenia, že pôvodný defekt bol úspešne odstránený. Takéto testovanie sa nazýva konfirmačné testovanie. Ladenie (lokalizácia a oprava defektu) je aktivita vývoja, nie aktivita testovania.

Regresné testovanie je opakované testovanie už testovaného programu po modifikácii s cieľom nájsť všetky defekty, ktoré boli zanesené alebo objavené ako dôsledok zmeny (zmien). Tieto defekty sa môžu nachádzať v testovanom softvéri alebo v inom súvisiacom alebo nesúvisiacom softvérovom komponente. Regresné testovanie sa vykonáva, keď je zmenený softvér alebo jeho prostredie. Rozsah regresného testovania je odvodený od rizika nenájdenia defektov v softvéri, ktorý predtým fungoval.

Testy by mali byť opakovateľné, ak sa majú používať v konfirmačnom testovaní a pomáhať regresnému testovaniu.

Regresné testovanie môže byť vykonávané na všetkých úrovniach testovania a zahŕňa funkcionálne, nefunkcionálne a štrukturálne testovanie. Sady regresných testov sú spustené veľa krát a všeobecne sa rozvíjajú pomaly, preto je regresné testovanie silným kandidátom pre automatizáciu.

2.4 Testovanie údržby (Z2)

15 minút

Základné výrazy

Dopadová analýza, testovanie údržby.

Pozadie

Raz vyvinutý softvérový systém je často v prevádzke roky až desaťročia. Počas tohto času sú systém, jeho konfiguračné dáta, alebo jeho prostredie často opravované, menené alebo rozširované. Plánovanie releasov v predstihu je rozhodujúce pre úspešné testovanie údržby. Musí sa rozlišovať medzi plánovanými releasmi a urgentnými opravami. Testovanie údržby je vykonávané na existujúcom produkčnom systéme a je vyvolané modifikáciami, migráciou alebo vyradením (z používania) softvéru alebo systému.

Modifikácie zahŕňajú plánované zlepšujúce zmeny (napr. založené na vydaní softvéru (release)), korigujúce a naliehavé núdzové zmeny, zmeny prostredia ako napríklad plánovaný upgrade operačného systému alebo databázy, plánované aktualizácie krabicového softvéru, alebo záplaty opravujúce novo odkryté alebo objavené riziká operačného systému.

Testovanie údržby pri migrácii (napr. z jednej platformy na inú), by malo zahŕňať prevádzkové testy nového prostredia, ako aj zmeneného softvéru. Testovanie migrácie (testovanie konverzie) je taktiež potrebné v prípadoch, keď budú do udržiavaného systému migrované dáta z inej aplikácie.

Testovanie údržby pri vyradení systému z používania môže zahŕňať testovanie dátovej migrácie alebo archivovania, v prípade, že sú požadované dlhé doby uchovávaní dát.

Navyše okrem testovania toho, čo bolo zmenené, zahŕňa testovanie údržby aj regresné testovanie tých častí softvéru, ktoré neboli zmenené. Rozsah testovania údržby súvisí s rizikom vyplývajúcim zo zmeny, s veľkosťou existujúceho systému a s veľkosťou zmeny. V závislosti na zmenách môže byť testovanie údržby vykonané na niektorých alebo na všetkých úrovniach testovania pre niektoré alebo všetky typy testov.

Určenie, ako môže byť existujúci systém ovplyvnený zmenami, sa nazýva dopadová analýza. Používa sa s cieľom pomôcť rozhodnúť sa, do akej miery vykonať regresné testovanie. Dopadová analýza môže byť použitá k určeniu regresnej testovacej sady.

Testovanie údržby môže byť náročné v prípade neaktuálnej alebo chýbajúcej špecifikácie, alebo ak nie sú k dispozícii testerí so znalosťou danej oblasti.

Referencie

- 2.1.3 CMMI, Craig, 2002, Hetzel, 1988, IEEE 12207
- 2.2 Hetzel, 1988
- 2.2.4 Copeland, 2004, Myers, 1979
- 2.3.1 Beizer, 1990, Black, 2001, Copeland, 2004
- 2.3.2 Black, 2001, ISO 9126
- 2.3.3 Beizer, 1990, Copeland, 2004, Hetzel, 1988
- 2.3.4 Hetzel, 1988, IEEE STD 829:1998
- 2.4 Black, 2001, Craig, 2002, Hetzel, 1988, IEEE STD 829:1998

3. Statické techniky (Z2)

60 minút

Študijné ciele pre statické techniky

Tieto ciele určujú, čo budete vedieť po dokončení každého modulu.

3.1 Statické techniky a proces testovania (Z2)

ŠC-3.1.1 Rozpoznať softvérové pracovné produkty, ktoré môžu byť prešetrené rôznymi statickými technikami. (Z1)

ŠC-3.1.2 Popísať dôležitosť a význam zvažovania statických techník pre hodnotenie softvérových pracovných produktov. (Z2)

ŠC-3.1.3 Vysvetliť rozdiel medzi statickými a dynamickými technikami s ohľadom na ciele, typy defektov, ktoré majú byť nájdené a rolu týchto techník v životnom cykle softvéru. (Z2)

3.2 Revízny proces (Z2)

ŠC-3.2.1 Zapamätať si aktivity, roly a zodpovednosti typickej formálnej revízie. (Z1)

ŠC-3.2.2 Vysvetliť rozdiely medzi rôznymi typmi revízií: neformálna revízia, technická revízia, walkthrough a inšpekcia. (Z2)

ŠC-3.2.3 Vysvetliť faktory úspešného vykonania revízií. (Z2)

3.3 Statická analýza s použitím nástrojov (Z2)

ŠC-3.3.1 Spomenúť si na typické defekty a omyly identifikované statickou analýzou a porovnať ich s revíziami a dynamickým testovaním. (Z1)

ŠC-3.3.2 Popísať za použitia príkladov zvyčajné prínosy statickej analýzy. (Z2)

ŠC-3.3.3 Vymenovať zvyčajné defekty v kóde a návrhu, ktoré môžu byť odhalené nástrojmi pre statickú analýzu. (Z1)

3.1 *Statické techniky a proces testovania (Z2)*

15 minút

Základné výrazy

Dynamické testovanie, statické testovanie.

Pozadie

Na rozdiel od dynamického testovania, ktoré vyžaduje spúšťanie softvéru, statické testovacie techniky sa spoliehajú na manuálne preskúmanie (revidovanie) a automatizovanú analýzu (statickú analýzu) kódu alebo inej projektovej dokumentácie bez vykonania kódu.

Revízie sú spôsobom testovania softvérových pracovných produktov (vrátane kódu) a môžu sa vykonať oveľa skôr ako dynamické testy. Defekty nájdené počas revízií v skorých fázach životného cyklu (napr. defekty nájdené v požiadavkách) sú často odstrániteľné oveľa lacnejšie ako tie, ktoré sú nájdené bežiacimi testami na vykonanom kóde.

Celá revízia môže byť vykonaná ako manuálna aktivita, ale existujú aj podporné nástroje. Hlavnou manuálnou aktivitou je preskúmanie pracovného produktu a zaznamenanie poznámok o ňom. Každý softvérový pracovný produkt môže byť revidovaný, vrátane špecifikácií požiadaviek, špecifikácií návrhu, kódu, testovacích plánov, testovacích špecifikácií, testovacích prípadov, testovacích skriptov, užívateľských príručiek alebo webových stránok.

Prínosy revízií zahŕňajú skorú detekciu a opravu defektov, vylepšenie produktivity vývoja, skrátený časový interval pre vývoj, znížené náklady a čas na testovanie, znížené náklady na životný cyklus softvéru, menej defektov a kvalitnejšiu komunikáciu. Revízie môžu nájsť pozabudnuté veci, napríklad v požiadavkách, ktoré by sa len s malou pravdepodobnosťou našli v rámci dynamického testovania.

Revízie, statická analýza a dynamické testovanie majú rovnaký cieľ – identifikáciu defektov. Navzájom sa dopĺňajú - rôzne techniky môžu nájsť rozdielne typy defektov efektívnejšie a účinnejšie. V porovnaní s dynamickým testovaním, statické techniky nájdu skôr príčiny zlyhaní (defekty), než samotné zlyhania.

Typické defekty, ktoré sú počas revidovania jednoduchšie odhaliteľné ako v dynamickom testovaní zahŕňajú: rozdiely oproti štandardom, defekty v požiadavkách, defekty v návrhu, nedostatočná udržiavateľnosť a nesprávne špecifikácie rozhrania.

3.2 Revízny proces (Z2)

25 minút

Základné výrazy

Vstupné kritérium, formálna revízia, neformálna revízia, inšpekcia, metrika, moderátor, odborná revízia, revidujúci, zapisovateľ, technická revízia, walkthrough.

Pozadie

Rôzne typy revízií sa líšia od neformálnych, ktoré sa vyznačujú tým, že nemajú spísané inštrukcie pre revidujúcich, až po systematické, ktoré sa vyznačujú tímovou účasťou, dokumentovanými výsledkami revízie a dokumentovanými postupmi pre vykonávanie revízie. Formálnosť revízneho procesu záleží od faktorov ako sú napríklad zrelosť procesu vývoja, právnych alebo regulačných požiadaviek, alebo potreby auditovacích záznamov.

Spôsob, akým je revízia vykonaná, závisí na dohodnutých cieľoch revízie (napr. nájsť defekty, získať porozumenie, vzdelávať testerov a nových členov tímu, alebo diskutovať a rozhodovať konsenzom).

3.2.1 Aktivity formálnej revízie (Z1)

Typická formálna revízia má nasledujúce hlavné aktivity:

1. Plánovanie:
 - definícia revíznych kritérií,
 - výber osôb,
 - alokácia rolí,
 - definícia vstupných a výstupných kritérií pri formálnejších typoch revízií (napr. inšpekcia),
 - výber častí dokumentov, na ktoré bude revízia zameraná,
 - preverenie vstupných kritérií (pri formálnejších typoch revízií).
2. Kick-off:
 - distribúcia dokumentov,
 - vysvetlenie cieľov, procesu a dokumentov účastníkom.
3. Individuálna príprava:
 - príprava na revízne stretnutie formou revízie dokumentu(ov),
 - zaznamenanie potenciálnych defektov, otázok a pripomienok.
4. Prešetrenie/vyhodnotenie/zaznamenanie výsledkov (revízne stretnutie):
 - diskutovanie alebo zaznamenávanie, s dokumentovanými výsledkami alebo zápisom (pri formálnejších typoch revízií),
 - zaznamenávanie defektov, dávanie odporúčaní ohľadne toho, čo robiť s defektami, robenie rozhodnutí ohľadne defektov,
 - prešetrenie/vyhodnotenie a zaznamenávanie problémov počas akéhokoľvek fyzického stretnutia alebo sledovanie akejkoľvek skupinovej elektronickej komunikácie.
5. Prepracovanie:
 - oprava nájdených defektov (zvyčajne vykonávaná autorom),
 - zaznamenanie aktualizovaného stavu defektov (vo formálnych revíziách).
6. Následné kroky:
 - preverenie, že o defekty bolo riadne postarané,
 - zbieranie metrik,
 - kontrola výstupných kritérií (pri formálnejších typoch revízií).

3.2.2 Roly a zodpovednosti (Z1)

Typická formálna revízia zahŕňa doleuvedené roly:

- Manažér: rozhoduje o vykonávaní revízií, alokuje čas v projektových harmonogramoch a určuje, či boli ciele revízie splnené.
- Moderátor: osoba, ktorá vedie revíziu dokumentu alebo sady dokumentov, čo zahŕňa vedenie plánovania revízie, priebehu stretnutia a následných krokov po stretnutí. V prípade potreby, môže moderátor robiť sprostredkovateľa medzi rôznymi pohľadmi na vec a je často osobou, od ktorej závisí úspech revízie.
- Autor: osoba s hlavnou zodpovednosťou za revidovaný(é) dokument(y).
- Revidujúci: jednotlivci so špecifickým technickým alebo obchodným pozadím (tiež nazývaní kontrolóri alebo inšpektori), ktorí, po nutnej príprave, identifikujú a popíšu zistenia (napr. defekty) v revidovanom produkte. Revidujúci by mali byť vybraní tak, aby reprezentovali rôzne pohľady a roly v procese revízie a mali by sa zúčastniť na akomkoľvek revíznom stretnutí.
- Zapisovateľ (alebo zaznamenávateľ): dokumentuje všetky zásadné otázky, problémy a otvorené body, ktoré boli identifikované počas stretnutia.

Prezeranie softvérových produktov alebo súvisiacich precovných produktov z rôznych perspektív a používanie kontrolných zoznamov môžu zvýšiť efektívnosť a účinnosť revízie. Napríklad kontrolný zoznam z pohľadu užívateľa, údržby, testera alebo operátora, alebo kontrolný zoznam typických problémov s požiadavkami môže napomôcť odhaliť dosiaľ nedetekované problémy.

3.2.3 Typy revízií (Z2)

Jeden softvérový produkt, alebo súvisiaci pracovný produkt môže byť predmetom viacerých revízií. V prípade, že je použitých viacero typov revízií, ich poradie môže byť rôzne. Napríklad neformálna revízia môže byť uskutočnená pred technickou revíziou, alebo inšpekcia špecifikácie požiadaviek môže byť vykonaná pred walkthrough so zákazníkmi. Hlavné charakteristiky, možnosti a účely bežných typov revízií sú:

Neformálna revízia

- žiadny formálny proces;
- môže mať formu programovania v pároch alebo formu revízie návrhu a kódu technickým vedúcim;
- výsledky môžu byť dokumentované;
- jej užitočnosť je rôzna v závislosti na revidujúcom;
- hlavný účel: lacný spôsob ako získať aspoň nejaký prínos.

Walkthrough

- stretnutie vedené autorom;
- môže mať formu scenárov, skúšok "nanečisto", účasti skupiny kolegov;
- sedenia s otvoreným koncom
 - voliteľná príprava revidujúcich pred stretnutím,
 - voliteľná príprava reportu revízie zahrňujúca zoznam zistení;
- voliteľný zapisovateľ (ktorý nie je autorom);
- môže sa v praxi vyskytovať od relatívne neformálneho až po veľmi formálne;
- hlavné účely: učenie, získanie porozumenia, nájdenie defektov.

Technická revízia

- dokumentovaná, definovaný proces odhaľovania defektov, ktorý zahŕňa kolegov a technických expertov s voliteľnou účasťou manažmentu;
- môže byť uskutočnená ako revízia kolegami na rovnakej úrovni bez účasti manažmentu;
- ideálne vedená zaškoleným moderátorom (nie autorom);
- príprava revidujúcich pred stretnutím;
- voliteľné použitie kontrolných zoznamov;

- príprava revízneho reportu, ktorý zahŕňa zoznam zistení, verdikt či softvérový produkt spĺňa svoje požiadavky a tam kde je to vhodné, odporúčania vzťahujúce sa k týmto zisteniam;
- môže sa v praxi vyskytovať od relatívne neformálnej až po veľmi formálnu;
- hlavné účely: diskutovanie, robenie rozhodnutí, vyhodnocovanie alternatív, nachádzanie defektov, riešenie technických problémov a preverovanie súladu so špecifikáciami, plánmi, smernicami a štandardmi.

Inšpekcia

- vedená zaškoleným moderátorom (nie autorom);
- zvyčajne vykonaná ako preskúmanie kolegami;
- definované roly;
- zahŕňa zber metrik;
- formálny proces založený na pravidlách a kontrolných zoznamoch
- špecifikované vstupné a výstupné kritériá pre akceptáciu softvérového produktu;
- príprava pred stretnutím;
- report inšpekcie zahrnujúci zoznam zistení;
- formálny proces následných krokov (s voliteľnými komponentmi zlepšovania procesu);
- voliteľný čítajúci;
- hlavný účel: nájdenie defektov.

Walkthrough, technické revízie a inšpekcie môžu byť vykonané v rámci skupiny kolegov – napr. kolegami na rovnakej organizačnej úrovni. Tento typ revízie sa nazýva „revízia kolegom“.

3.2.4 Faktory úspechu pre revízie (Z2)

Faktory úspechu pre revízie zahŕňajú:

- Každá revízia má jasne preddefinované ciele.
- Revíziou sa zaoberajú vhodní ľudia s ohľadom na jej ciele.
- Testerí sú považovaní za hodnotných revidujúcich, ktorí prispievajú k revízii a ktorí sa taktiež učia o produkte, čo im umožňuje skôr pripraviť testy.
- Nájdené defekty sú vítané a sú vyjadrené objektívne.
- Ľudské spory a psychologické aspekty sú zvládnuté (napr. sa stali pozitívnou skúsenosťou pre autora).
- Revízia je vykonaná v atmosfére dôvery, výsledok nebude použitý na hodnotenie zúčastnených.
- Revízne techniky sú aplikované vhodne za účelom dosiahnutia cieľov a s ohľadom na typ a úroveň softvérových pracovných produktov a revidujúcich.
- Kontrolné zoznamy alebo roly sú používané, ak je to vhodné za účelom zvýšenia efektívnosti identifikácie defektov.
- Je zabezpečené školenie revíznych techník, obzvlášť pri formálnejších technikách ako napríklad inšpekcia.
- Manažment podporuje správny proces revízie (napr. vyčlenením dostatočného času na revízne aktivity v rámci projektového harmonogramu).
- Kládne sa dôraz na učenie a zlepšovanie procesu.

3.3 *Statická analýza s použitím nástrojov (Z2)*

20 minút

Základné výrazy

Kompilátor, komplexita, riadiaci tok, dátový tok, statická analýza.

Pozadie

Cieľom statickej analýzy je nájsť defekty v zdrojovom kóde softvéru a v softvérových modeloch. Statická analýza je vykonávaná bez vlastného spúšťania programu skúmaného nástrojom, zatiaľ čo dynamické testovanie spúšťa kód programu. Statická analýza dokáže objaviť defekty, ktoré sú ťažko odhaliteľné v dynamickom testovaní. Rovnako ako pri revíziách, statická analýza nájde skôr defekty ako zlyhania. Nástroje pre statickú analýzu analyzujú programový kód (napr. riadiaci tok a dátový tok), ako aj vygenerované výstupy ako HTML a XML.

Význam statickej analýzy je:

- Skorá detekcia defektov ešte pred vykonaním testu.
- Skoré varovanie o podozrivých aspektoch kódu alebo návrhu vypočítaním metrík ako je napríklad vysoká miera komplexnosti.
- Identifikácia defektov, ktoré by nebolo jednoduché odhaliť dynamickým testovaním.
- Detekcia závislostí a nekonzistencií v softvérovom modeli, ako napríklad prepojenia.
- Zlepšená udržiavateľnosť kódu a návrhu.
- Prevencia defektov v prípade, že došlo k ponaučeniu z vývoja.

Typické defekty objavené nástrojmi na statickú analýzu zahŕňajú:

- odkaz na premennú bez definovanej hodnoty;
- nekonzistentné rozhrania medzi modulmi a komponentmi;
- premenné, ktoré nie sú použité alebo sú nesprávne deklarované;
- nedosiahnuteľný (mŕtvy) kód;
- chýbajúca a chybná logika (potenciálne nekonečné slučky);
- príliš komplikované konštrukty;
- porušenia štandardov programovania;
- bezpečnostné nedostatky;
- porušenia syntaxe kódu a softvérových modelov.

Nástroje pre statickú analýzu sú zvyčajne používané vývojármi (preverenie voči preddefinovaným pravidlám alebo štandardom programovania) pred a počas testovania komponentov a integračného testovania, alebo pri ukladaní kódu do nástroja pre konfiguračný manažment a návrhármi počas modelovania softvéru. Nástroje pre statickú analýzu môžu produkovať veľké množstvo varovných hlásení, ktoré je potrebné dobre riadiť, aby bol nástroj využitý čo najefektívnejšie.

Kompilátory môžu poskytovať čiastočnú podporu pre statickú analýzu, vrátane výpočtu metrík.

Referencie

- 3.2 IEEE 1028
- 3.2.2 Gilb, 1993, van Veenendaal, 2004
- 3.2.4 Gilb, 1993, IEEE 1028
- 3.3 van Veenendaal, 2004

4. Techniky tvorby testov (Z4)

285 minút

Študijné ciele pre techniky tvorby testov

Tieto ciele určujú, čo budete vedieť po dokončení každého modulu.

4.1 Proces vývoja testu (Z3)

- ŠC-4.1.1 Rozlišovať medzi špecifikáciou návrhu testu, špecifikáciou testovacieho prípadu a špecifikáciou testovacej procedúry. (Z2)
- ŠC-4.1.2 Porovnať pojmy testovacia podmienka, testovací prípad a testovacia procedúra. (Z2)
- ŠC-4.1.3 Vyhodnotiť kvalitu testovacích prípadov s ohľadom na jasnú sledovateľnosť k požiadavkám a očakávané výsledky. (Z2)
- ŠC-4.1.4 Previesť testovacie prípady do dobre štruktúrovanej špecifikácie testovacej procedúry na úrovni detailu zodpovedajúcej znalostiam testerov. (Z3)

4.2 Kategórie techník návrhu testov (Z2)

- ŠC-4.2.1 Zapamätať si dôvody, prečo sú užitočné techniky návrhu testov založené na špecifikácii (čierna skrinka) a prístupy založené na štruktúre (biela skrinka) a pre každú z týchto skupín vymenovať bežné techniky. (Z1)
- ŠC-4.2.2 Vysvetliť charakteristiky, to čo majú spoločné a rozdiely medzi testovaním založenom na špecifikácii, štruktúre a skúsenosti. (Z2)

4.3 Techniky založené na špecifikácii alebo techniky čiernej skrinky (Z3)

- ŠC-4.3.1 Napísať testovacie prípady z existujúcich softvérových modelov s použitím rozdelenia ekvivalencie, analýzy hraničných hodnôt, rozhodovacích tabuliek a diagramov/tabuliek prechodu stavov. (Z3)
- ŠC-4.3.2 Vysvetliť hlavný zámer každej zo štyroch testovacích techník, aká úroveň a typ testovania môže techniku využiť a akým spôsobom môže byť merané pokrytie. (Z2)
- ŠC-4.3.3 Vysvetliť koncept testovania prípadov použitia a jeho prínosy. (Z2)

4.4 Techniky založené na štruktúre alebo techniky bielej skrinky (Z4)

- ŠC-4.4.1 Popísať koncept a hodnotu pokrytia kódu. (Z2)
- ŠC-4.4.2 Vysvetliť koncepty pokrytia príkazov a rozhodnutí a uviesť dôvody, prečo sa tieto koncepty môžu použiť aj na iných úrovniach testovania ako testovanie komponentov (napr. na biznis procedúrach na úrovni systému). (Z2)
- ŠC-4.4.3 Napísať testovacie prípady pre dané riadiace toky s použitím techník návrhu testov príkazov a rozhodovaní. (Z3)
- ŠC-4.4.4 Ohodnotiť pokrytie príkazov a rozhodovaní z hľadiska úplnosti s ohľadom na definované výstupné kritériá. (Z4)

4.5 Techniky založené na skúsenosti (Z2)

- ŠC-4.5.1 Zapamätať si príčiny pre písanie testovacích prípadov založené na intuícii, skúsenosti a znalosti o obvyklých defektoch. (Z1)
- ŠC-4.5.2 Porovnať techniky založené na skúsenosti s technikami testovania založenými na špecifikácii. (Z2)

4.6 Výber testovacích techník (Z2)

ŠC-4.6.1 Klasifikovať techniky návrhu testov podľa ich vhodnosti pre daný kontext, základ testovania, súvisiace modely a softvérové charakteristiky. (Z2)

4.1 *Proces vývoja testu (Z3)*

15 minút

Základné výrazy

Špecifikácia testovacieho prípadu, návrh testu, harmonogram vykonania testov, špecifikácia testovacej procedúry, testovací skript, sledovateľnosť.

Pozadie

Proces vývoja testu popisovaný v tejto kapitole sa môže vykonávať rôznymi spôsobmi, od veľmi neformálneho so stručnou alebo žiadnou dokumentáciou až po veľmi formálny (ako je to popísané nižšie). Stupeň formálnosti závisí od kontextu testovania, vrátane zrelosti testovania a vývojového procesu, časových obmedzení, bezpečnostných alebo regulačných požiadaviek a zapojených ľudí.

Počas analýzy testovania je analyzovaná dokumentácia základu testovania s cieľom určiť, čo sa bude testovať, t.j. identifikovať testovacie podmienky. Testovacia podmienka je definovaná ako položka alebo udalosť, ktorá môže byť verifikovaná jedným alebo viacerými testovacími prípadmi (napr. funkcia, transakcia, kvalitatívna charakteristika alebo štrukturálny element).

Zavedenie sledovateľnosti od testovacích podmienok späť k špecifikáciám a požiadavkám umožňuje nie len účinnú dopadovú analýzu, keď sa požiadavky menia, ale aj určenie pokrytia požiadaviek pre sadu testov. Počas analýzy testovania sa implementuje detailný prístup k testovaniu za účelom výberu techník návrhu testov, ktoré budú (okrem iných hľadísk) použité na základe identifikovaných rizík (pozri kapitolu 5 pre viac informácií o analýze rizík).

Počas návrhu testov sú vytvárané a špecifikované testovacie prípady a testovacie dáta. Testovací prípad pozostáva zo sady vstupných hodnôt, predpokladov vykonania, očakávaných výsledkov a výstupných podmienok po vykonaní. Je vyvinutý tak, aby pokryl určitý(é) testovací(ie) ciele alebo testovaci(u) podmienku(y). Štandard pre dokumentáciu testovania softvéru (IEEE STD 829-1998) popisuje obsah špecifikácií návrhu testov (obsahujúcich testovacie podmienky) a špecifikácií testovacích prípadov.

Očakávané výsledky by mali byť vytvorené ako časť špecifikácie testovacieho prípadu a mali by zahŕňať výstupy, zmeny dát a stavov a akékoľvek ďalšie následky testu. Ak nebudú očakávané výsledky definované, môže byť prijateľný ale chybný výsledok, interpretovaný ako správny. Očakávané výsledky by mali byť v ideálnom prípade definované skôr ako sa začnú vykonávať testy.

Počas implementácie testu sú vyvíjané a implementované testovacie prípady, je určovaná ich priorita a sú organizované do špecifikácie testovacích procedúr (IEEE STD 829-1998). Testovacia procedúra špecifikuje poradie činností pre vykonanie testu. Ak sú testy spustené využitím nástroja na vykonanie testu, poradie činností je špecifikované v testovacom skripte (čo je automatizovaná testovacia procedúra).

Rôzne testovacie procedúry a automatizované testovacie skripty sú potom sformované do harmonogramu vykonávania testov. Ten definuje poradie, v ktorom sú vykonávané rôzne testovacie procedúry a prípadne automatizované testovacie skripty. Harmonogram vykonania testov berie do úvahy také faktory, ako sú regresné testy, stanovenie priorít, technické a logické závislosti.

4.2 Kategórie techník návrhu testov (Z2)

15 minút

Základné výrazy

Technika návrhu testov čiernej skrinky, technika návrhu testov založená na skúsenostiach, technika návrhu testov, technika návrhu testov bielej skrinky.

Pozadie

Účelom techniky návrhu testov je identifikovať testovacie podmienky, testovacie prípady a testovacie dáta.

Tradične sa testovacie techniky označujú ako techniky čiernej skrinky alebo ako techniky bielej skrinky. Techniky návrhu testov čiernej skrinky (nazývané taktiež techniky založené na špecifikácii) slúžia na odvodenie a výber testovacích podmienok, testovacích prípadov, alebo testovacích dát. Tieto techniky vychádzajú z analýzy dokumentácie základu testovania a zahŕňajú funkcionálne aj nefunkcionálne testovanie. Testovanie čiernej skrinky podľa definície nepoužíva žiadnu informáciu o vnútornej štruktúre komponentu alebo systému, ktorý má byť testovaný. Techniky návrhu testov bielej skrinky (taktiež nazývané štruktúrne techniky alebo techniky založené na štruktúre) sú založené na analýze štruktúry komponentu alebo systému. Testovanie čiernej skrinky a testovanie bielej skrinky môže byť taktiež kombinované s technikami založenými na skúsenostiach za účelom zužitkovania skúseností vývojárov, testerov a užívateľov k určeniu toho, čo má byť testované.

Niektoré techniky sa zaraďujú jasne do jednej kategórie; iné obsahujú prvky viac ako jednej kategórie.

Táto učebná osnova označuje techniky návrhu testov založených na špecifikácii ako techniky čiernej skrinky a techniky návrhu testov založených na štruktúre ako techniky bielej skrinky. Pokrýva navyše ešte techniky návrhu testov založených na skúsenostiach.

Spoločné vlastnosti techník návrhu testov založených na špecifikácii zahŕňajú:

- Modely, formálne alebo neformálne, sa používajú pre špecifikovanie riešeného problému, softvéru alebo jeho komponentov.
- Z týchto modelov sa môžu systematicky odvodzovať testovacie prípady.

Spoločné vlastnosti techník návrhu testov založených na štruktúre zahŕňajú:

- Pri odvodzovaní testovacích prípadov sa používa informácia o tom, ako je softvér vytvorený (napr. kód a detailné informácie o návrhu).
- Rozsah pokrytia softvéru môže byť meraný pre existujúce testovacie prípady a ďalšie testovacie prípady môžu byť systematicky odvodené za účelom zvýšenia pokrytia.

Spoločné vlastnosti techník návrhu testov založených na skúsenostiach zahŕňajú:

- Pri odvodzovaní testovacích prípadov sa používajú znalosti a skúsenosti ľudí.
- Jedným zdrojom informácií sú znalosti testerov, vývojárov, užívateľov a ďalších kľúčových osôb o softvéri, jeho použití a prostredí.
- Ďalším zdrojom informácií sú znalosti o pravdepodobných defektoch a ich distribúcii.

4.3 *Techniky založené na špecifikácii alebo techniky čiernej skrinky (Z3)*

150 minút

Základné výrazy

Analýza hraničných hodnôt, testovanie rozhodovacej tabuľky, rozdelenie ekvivalencie, testovanie prechodu stavov, testovanie prípadov použitia.

4.3.1 Rozdelenie ekvivalencie (Z3)

Pri rozdelení ekvivalencie sú vstupy softvéru alebo systému rozdelené do skupín, v ktorých sa očakáva rovnaké správanie, takže sú pravdepodobne spracované rovnakým spôsobom. Sekcie ekvivalencie (alebo triedy) môžu byť nájdené pre platné dáta, t.j. hodnoty, ktoré by mali byť akceptované, ako aj pre neplatné dáta, t.j. hodnoty ktoré by mali byť zamietnuté. Sekcie môžu byť identifikované tiež pre výstupy, vnútorné hodnoty, časovo súvisiace hodnoty (napr. pred alebo po udalosti) a pre parametre rozhraní (napr. testované integrované komponenty počas integračného testovania). Testy môžu byť navrhnuté aby pokryli všetky platné a neplatné sekcie. Rozdelenie ekvivalencie je aplikovateľné na všetkých úrovniach testovania.

Rozdelenie ekvivalencie môže byť použité na dosiahnutie cieľov pokrytia vstupu a výstupu. To môže byť uplatnené pri manuálnom vstupe, vstupe cez rozhrania do systému alebo parametroch rozhrania v integračnom testovaní.

4.3.2 Analýza hraničných hodnôt (Z3)

Správanie na okraji každého rozdelenia ekvivalencie môže byť s väčšou pravdepodobnosťou nesprávne než je správanie vnútri sekcie, preto sú hranice oblasťou, kde testovanie môže pravdepodobne nájsť defekty. Maximálne a minimálne hodnoty sekcie predstavujú jej hraničné hodnoty. Hraničná hodnota pre platnú sekciu je platná hraničná hodnota; hranica neplatnej sekcie je neplatná hraničná hodnota. Testy môžu byť navrhnuté tak, aby pokryli oboje, platné aj neplatné hraničné hodnoty. Počas návrhu testovacích prípadov je vybraný test pre každú hraničnú hodnotu.

Analýza hraničných hodnôt môže byť aplikovaná na všetkých úrovniach testovania. Je relatívne jednoducho aplikovateľná a jej schopnosť odhalenia defektov je vysoká. Detailné špecifikácie sú užitočné pri určovaní zaujímavých hraníc.

Táto technika je často považovaná za rozšírenie rozdelenia ekvivalencie alebo iných techník návrhu testov čiernej skrinky. Môže byť použitá na triedach ekvivalencie pre užívateľský vstup cez obrazovku, ako aj napríklad pre časové rozsahy (napr. čas vypršania, požiadavky na rýchlosť transakcií), alebo rozsahy tabuliek (napr. veľkosť tabuľky je 256*256).

4.3.3 Testovanie rozhodovacích tabuliek (Z3)

Rozhodovacie tabuľky sú dobrým spôsobom ako zachytiť požiadavky systému, ktoré obsahujú logické podmienky a zdokumentovať návrh interného systému. Môžu byť použité pre nahrávanie zložitých biznis pravidiel, ktoré má systém implementovať. Pri vytváraní rozhodovacích tabuliek je analyzovaná špecifikácia a identifikované sú podmienky a činnosti systému. Vstupné podmienky a činnosti sú najčastejšie určené takým spôsobom, že nadobúdajú hodnoty Pravda alebo Nepravda (typu Boolean). Rozhodovacia tabuľka obsahuje spúšťacie podmienky, často kombinácie Pravda a Nepravda pre všetky vstupné podmienky a výsledné činnosti pre každú kombináciu podmienok. Každý stípec tabuľky zodpovedá biznis pravidlu, ktoré definuje unikátnu kombináciu podmienok a ktoré vedú k vykonaniu činností súvisiacich s týmto pravidlom. Štandard pokrytia, zvyčajne používaný spolu s testovaním rozhodovacej tabuľky, je mať najmenej jeden test na každý stípec v tabuľke, čo zvyčajne zahŕňa pokrytie všetkých kombinácií spúšťacích podmienok.

Výhodou testovania rozhodovacích tabuliek je vytvorenie kombinácií podmienok, ktoré by inak nemuseli byť preskúšané počas testovania. Táto technika môže byť aplikovaná na všetky situácie, keď je činnosť softvéru závislá na niekoľkých logických rozhodnutiach.

4.3.4 Testovanie prechodu stavov (Z3)

Systém môže dávať rôznu odpoveď v závislosti na aktuálnych podmienkach alebo predchádzajúcej histórii (jeho stave). V tomto prípade môže byť tento aspekt systému reprezentovaný ako diagram prechodu stavov. Dovoľuje testerovi chápať softvér v zmysle jeho stavov, prechodov medzi stavmi, vstupov alebo udalostí, ktoré spúšťajú zmeny stavov (prechody) a činností, ktoré môžu z týchto prechodov vyplývať. Stav systému alebo objektu, ktorý je predmetom testovania sú samostatné, identifikovateľné a je ich konečné množstvo.

Stavová tabuľka ukazuje vzťah medzi stavmi a vstupmi a môže zvýrazniť možné prechody, ktoré sú neplatné.

Testy môžu byť navrhnuté s cieľom pokryť typické sekvencie stavov, pokryť každý stav, vykonať každý prechod, vykonať špecifické sekvencie prechodov, alebo testovať neplatné prechody stavov.

Testovanie prechodu stavov je veľmi používané v oblasti vnorených (embedded) softvérových produktov a v technickej automatizácii vo všeobecnosti. Avšak táto technika je taktiež vhodná pre modelovanie biznis objektov, ktoré majú špecifické stavy alebo testovanie tokov obrazoviek (napr. pre Internetové aplikácie alebo biznis scenáre).

4.3.5 Testovanie prípadov použitia (Z2)

Testy môžu byť odvodené z prípadov použitia. Prípady použitia popisuje interakcie medzi aktérmi (užívateľmi alebo systémami), ktoré produkujú výsledok, ktorý má hodnotu pre užívateľa systému alebo zákazníka. Prípady použitia môžu byť popísané na abstraktnej úrovni (obchodný prípad použitia, bez popisu technológie, na úrovni obchodného procesu), alebo na úrovni systému (systémový prípad použitia na úrovni funkcionality systému). Každý prípad použitia má predpoklady, ktoré musia byť splnené, aby prípad použitia úspešne fungoval. Každý prípad použitia je ohraničený výstupnými podmienkami, ktoré sú pozorovateľnými výsledkami a finálnym stavom systému po tom, ako je prípad použitia ukončený. Prípady použitia má zvyčajne základný (t.j. najpravdepodobnejší) scenár a alternatívne scenáre.

Prípady použitia popisujú "procesné toky" cez systém založené na jeho pravdepodobnom reálnom použití, preto sú testovacie prípady odvodené z prípadov použitia najužitočnejšie pri odhaľovaní defektov v procesných tokoch počas skutočného používania systému. Prípady použitia sú veľmi užitočné pre navrhovanie akceptačných testov s účasťou zákazníka/užívateľa. Taktiež pomôžu odhaliť integračné defekty spôsobené interakciou a vzájomnou interferenciou rozličných komponentov, ktoré by testovanie jednotlivých komponentov neodhalilo. Navrhovanie testovacích prípadov z prípadov použitia môže byť kombinované s inými technikami testovania založených na špecifikácii.

4.4 *Techniky založené na štruktúre alebo techniky bielej skrinky (Z4)*

60 minút

Základné výrazy

Pokrytie kódu, pokrytie rozhodnutí, pokrytie príkazov, testovanie založené na štruktúre.

Pozadie

Testovanie založené na štruktúre/testovanie bielej skrinky je založené na identifikovanej štruktúre softvéru alebo systému, ako je zrejmé z nasledujúcich príkladov:

- Úroveň komponentu: štruktúra softvérového komponentu, t.j. príkazy, rozhodnutia, vetvy, alebo rovnobežne rozdielne cesty.
- Integrovaná úroveň: štruktúra môže byť strom volaní (diagram, v ktorom moduly volajú ďalšie moduly).
- Systémová úroveň: štruktúra môže byť štruktúra menu, biznis proces alebo štruktúra webovej stránky.

V tejto časti sa rozoberajú tri, s kódom súvisiace štruktúrne techniky návrhu testov pre pokrytie kódu založené na príkazoch, vetvách a rozhodnutiach. Pre testovanie rozhodovaní môže byť použitý diagram riadiacich tokov na vizualizáciu alternatív pre každé rozhodovanie.

4.4.1 Testovanie a pokrytie príkazov (Z4)

V testovaní komponentov predstavuje pokrytie príkazov stanovenie percenta vykonateľných príkazov, ktoré boli vykonané sadou testovacích prípadov. Technika testovania príkazov odvodzuje testovacie prípady s cieľom vykonať špecifické príkazy, spravidla za účelom zvýšenia pokrytia príkazov.

Pokrytie príkazov je určené podielom počtu vykonateľných príkazov, ktoré sú pokryté (navrhnutými alebo vykonanými) testovacími prípadmi a počtom všetkých vykonateľných príkazov v testovanom kóde.

4.4.2 Testovanie a pokrytie rozhodovaní (Z3)

Pokrytie rozhodovaní, vzťahujúce sa k testovaniu vetvenia, je stanovenie percenta výsledkov rozhodovania (napr. vetva Pravda a Nepravda IF príkazu), ktoré boli vykonané sadou testovacích prípadov. Technika testovania rozhodovaní odvodzuje testovacie prípady s cieľom vykonať špecifické výsledky rozhodovaní. Vetvy vytvárajú rozhodovacie body v kóde a zobrazujú prenos riadenia do iných oblastí v kóde.

Pokrytie rozhodovaní je určené podielom počtu všetkých výsledkov rozhodovaní, ktoré sú pokryté (navrhnutými alebo vykonanými) testovacími prípadmi a počtom všetkých možných výsledkov rozhodovaní v testovanom kóde.

Testovanie rozhodovaní je forma testovania riadiacich tokov, pretože sleduje špecifický tok riadenia cez rozhodovacie body. Pokrytie rozhodovaní je silnejšie ako pokrytie príkazov: 100% pokrytie rozhodovaní garantuje 100% pokrytie príkazov, ale nie naopak.

4.4.3 Ďalšie techniky založené na štruktúrach (Z1)

Existujú silnejšie úrovne pokrytia štruktúr okrem pokrytia rozhodovaní, napríklad pokrytie podmienok a pokrytie viacnásobných podmienok.

Certifikovaný tester

Učebná osnova pre základný stupeň



Koncept pokrytia môže byť aplikovaný taktiež na ďalšie úrovne testov. Napríklad na integračnej úrovni môže byť percento modulov, komponentov alebo tried, ktoré boli vykonané sadou testovacích prípadov vyjadrené ako pokrytie modulov, komponentov alebo tried.

Podpora nástrojmi je užitočná pre štruktúralne testovanie kódu.

4.5 Techniky založené na skúsenosti (Z2)

30 minút

Pojmy

Prieskumné testovanie, útok (na chyby).

Pozadie

Pri testovaní založenom na skúsenostiach sú testy odvádzané zo znalostí a intuície testerov a ich skúsenosti s podobnými aplikáciami a technológiami. Ak sa použijú na rozšírenie systematických techník, môžu byť tieto techniky užitočné pri identifikácii špeciálnych testov, ktoré je náročné zachytiť formálnymi technikami, najmä v prípade, ak sú aplikované následne po viac formálnych prístupoch. Avšak táto technika môže vykazovať značne premenlivé stupne efektívnosti, v závislosti od skúsenosti testerov.

Zvyčajne používanou technikou založenou na skúsenostiach je odhadovanie omylov. Testerí všeobecne predvídajú defekty na základe skúseností. Štrukturovaný prístup k technike odhadovania omylov znamená vypracovanie zoznamu možných defektov a návrhu testov, ktoré na tieto defekty budú útočiť. Tento systematický prístup sa volá útok na chyby. Tieto zoznamy defektov a zlyhaní môžu byť vytvorené na základe skúsenosti, dostupných dát o defektoch a zlyhaniach a na základe všeobecných znalostí o tom, prečo softvér zlyháva.

Prieskumné testovanie je súbežný návrh testov, vykonávanie testov, zaznamenávanie testov a učenie. Je založené na charte testovania obsahujúcej ciele testovania a vykonávané v časových rámcoch. Je to spôsob, ktorý je najužitočnejší tam, kde sú nedostatočné alebo neadekvátne špecifikácie a silný časový tlak, resp. sa používa s cieľom rozšíriť alebo doplniť iné, viac formálne testovanie. Môže slúžiť ako kontrola procesu testovania pomáhajúca zabezpečiť nájdenie najzávažnejších defektov.

4.6 Výber testovacích techník (Z2)

15 minút

Pojmy

Žiadne špecifické pojmy.

Pozadie

Rozhodnutie, ktoré techniky použiť, závisí od množstva faktorov, ako sú typ systému, regulačné štandardy, zákaznícke alebo zmluvné požiadavky, úroveň rizika, typ rizika, cieľ testovania, dostupná dokumentácia, znalosti testerov, čas a rozpočet, životný cyklus vývoja, modely prípadov použitia, predchádzajúce skúsenosti so zistenými typmi defektov.

Niektoré techniky sú lepšie aplikovateľné len na určité situácie a úrovne testovania; ďalšie sú aplikovateľné na všetky úrovne testovania.

Pri vytváraní testovacích prípadov zvyčajne testerí používajú kombináciu testovacích techník vrátane procesných techník a techník založených na pravidlách a dátach za účelom zaistenia dostatočného pokrytia objektu testovania.

Referencie

- 4.1 Craig, 2002, Hetzel, 1988, IEEE STD 829-1998
- 4.2 Beizer, 1990, Copeland, 2004
- 4.3.1 Copeland, 2004, Myers, 1979
- 4.3.2 Copeland, 2004, Myers, 1979
- 4.3.3 Beizer, 1990, Copeland, 2004
- 4.3.4 Beizer, 1990, Copeland, 2004
- 4.3.5 Copeland, 2004
- 4.4.3 Beizer, 1990, Copeland, 2004
- 4.5 Kaner, 2002
- 4.6 Beizer, 1990, Copeland, 2004

5. Manažment testovania (Z3)

170 minút

Študijné ciele pre manažment testovania

Tieto ciele určujú, čo budete vedieť po dokončení každého modulu.

5.1 Organizácia testovania (Z2)

- ŠC-5.1.1 Pochopiť dôležitosť nezávislého testovania. (Z1)
- ŠC-5.1.2 Vysvetliť výhody a nevýhody nezávislého testovania v organizácii. (Z2)
- ŠC-5.1.3 Pochopiť rôznych členov testovacieho tímu potrebných pre jeho zostavenie. (Z1)
- ŠC-5.1.4 Zapamätať si úlohy typického vedúceho testovania a testera. (Z1)

5.2 Plánovanie a odhadovanie testovania (Z3)

- ŠC-5.2.1 Pochopiť rôzne úrovne a ciele plánovania testovania. (Z1)
- ŠC-5.2.2 Zhrnúť účel a obsah testovacieho plánu, špecifikácie návrhu testov a dokumentov testovacích procedúr na základe Štandardu pre dokumentáciu testovania softvéru 'Standard for Software Test Documentation' (IEEE 829-1998). (Z2)
- ŠC-5.2.3 Rozlišovať medzi koncepčne odlišnými prístupmi k testovaniu, ako sú analytický, založený na modeloch, metodický, zohľadňujúci procesy/štandardy, dynamický/heuristický, konzultatívny a regresno-averzný. (Z2)
- ŠC-5.2.4 Rozlišovať medzi predmetom plánovania testovania pre systém a prípravou harmonogramu vykonania testov. (Z2)
- ŠC-5.2.5 Napísať harmonogram vykonania testov pre daný súbor testovacích prípadov s ohľadom na stanovenie priorít a technické a logické závislosti. (Z3)
- ŠC-5.2.6 Vymenovať aktivity prípravy a vykonávania testov, ktoré by mali byť zohľadnené počas plánovania testovania. (Z1)
- ŠC-5.2.7 Zapamätať si typické faktory, ktoré ovplyvňujú úsilie vo vzťahu k testovaniu. (Z1)
- ŠC-5.2.8 Odlíšiť dva konceptuálne rozdielne prístupy odhadovania: prístup založený na metrikách a expertný prístup. (Z2)
- ŠC-5.2.9 Pochopiť/zdôvodniť adekvátne vstupné a výstupné kritériá pre špecifické úrovne testovania a skupiny testovacích prípadov (napr. pre integračné testovanie, akceptačné testovanie alebo testovacie prípady pre testovanie použiteľnosti). (Z2)

5.3 Sledovanie a riadenie postupu testovania (Z2)

- ŠC-5.3.1 Zapamätať si bežné metriky používané pre monitorovanie prípravy a vykonávania testov. (Z1)
- ŠC-5.3.2 Vysvetliť a porovnať metriky testovania pre reportovanie z testovania a riadenie testovania (napr. nájdené a opravené defekty, úspešné a neúspešné testy) vzťahujúce sa k účelu a použitiu. (Z2)
- ŠC-5.3.3 Zhrnúť účel a obsah sumárneho reportu z testovania podľa Štandardu pre dokumentáciu testovania softvéru 'Standard for Software Test Documentation' (IEEE 829-1998). (Z2)

5.4 Konfiguračný manažment (Z2)

- ŠC-5.4.1 Zhrnúť, ako konfiguračný manažment podporuje testovanie. (Z2)

5.5 Riziko a testovanie (Z2)

- ŠC-5.5.1 Popísať riziko ako možný problém, ktorý by mohol ohroziť dosiahnutie cieľov jedného alebo viacerých kľúčových osôb na projekte. (Z2)

ŠC-5.5.2 Zapamätať si, že úroveň rizika je určená pravdepodobnosťou (výskytu) a dopadom (vzniknutej škody, ak riziko nastane). (Z1)

ŠC-5.5.3 Rozlíšiť projektové a produktové riziká. (Z2)

ŠC-5.5.4 Spoznať typické produktové a projektové riziká. (Z1)

ŠC-5.5.5 Popísať príkladmi, ako môže byť v plánovaní testovania použitá analýza a manažment rizík. (Z2)

5.6 Riadenie incidentov (Z3)

ŠC-5.6.1 Pochopiť obsah záznamu o incidente podľa Štandardu pre dokumentáciu testovania softvéru 'Standard for Software Test Documentation' (IEEE 829-1998). (Z1)

ŠC-5.6.2 Napísať záznam o incidente, ktorý obsahuje pozorovanie zlyhania počas testovania. (Z3)

5.1 Organizácia testovania (Z2)

30 minút

Základné výrazy

Tester, vedúci testovania, manažér testovania.

5.1.1 Organizácia testovania a nezávislosť (Z2)

Efektívnosť nachádzania defektov testovaním a revidovaním môže byť zvýšená použitím nezávislých testerov. Možnosti pre nezávislosť zahŕňa nasledovné:

- Žiadni nezávislí tester; vývojári testujú vlastný kód.
- Nezávislí tester v rámci vývojových tímov.
- Nezávislý testovací tím alebo skupina v rámci organizácie reportujúca projektovému alebo výkonnému manažmentu.
- Nezávislí tester z rovnakej biznis organizácie, alebo komunity užívateľov.
- Nezávislí špecialisti pre špecifické oblasti testovania ako napríklad tester pre použiteľnosť, bezpečnosť alebo certifikáciu (ktorí certifikujú softvérový produkt voči štandardom a predpisom).
- Nezávislí tester prenajatí alebo externí voči organizácii.

Pre veľké, zložité alebo z hľadiska bezpečnosti kritické projekty, je obvyčajne najlepšie mať početné úrovne testovania, s tým, že niektoré alebo všetky úrovne sú realizované nezávislými testerami. Vývoj môže na testovaní participovať, najmä na nižších úrovniach, ale jeho nedostatok objektivity často obmedzuje jeho efektívnosť. Nezávislí tester môžu mať právo požadovať a definovať testovacie procesy a pravidlá, ale mali by nadobúdať takéto roly vzťahujúce sa k procesu len za existencie jasného mandátu manažmentu.

Výhody nezávislosti zahŕňajú:

- Nezávislí tester vidia iné a rozdielne defekty a sú nezaujatí.
- Nezávislý tester môže verifikovať predpoklady, ktoré boli stanovené počas špecifikácie a implementácie systému.

Nevýhody zahŕňajú:

- Izoláciu od vývojového tímu (ak je tím úplne nezávislý).
- Nezávislí tester môžu byť úzkym hrdlom ako posledný kontrolný bod.
- Vývojári môžu stratiť zmysel pre zodpovednosť za kvalitu.
- Nezávislí tester môžu byť považovaní za úzke hrdlo, alebo obviňovaní za zdržania vo vydaní softvéru (release).

Testovacie úlohy môžu byť realizované ľuďmi v špecifickej testovacej role, alebo niekým v inej role, ako napríklad projektový manažér, manažér kvality, vývojár, expert v príslušnom biznise a odbore, človek z infraštruktúry alebo IT prevádzky.

5.1.2 Úlohy vedúceho testovania a testera (Z1)

Táto učebná osnova sa zaoberá dvomi testovacími rolami, vedúci testovania a tester. Činnosti a úlohy vykonávané ľuďmi v týchto dvoch rolách sú závislé na kontexte projektu a produktu, ľuďoch v daných rolách a organizácii.

Niekedy sa vedúci testovania nazýva manažér testovania alebo koordinátor testovania. Rola vedúceho testovania môže byť vykonávaná projektovým manažérom, manažérom vývoja, manažérom pre zabezpečenie kvality alebo manažérom testovacej skupiny. Vo väčších projektoch môžu existovať

dve pozície: vedúci testovania a manažér testovania. Spravidla vedúci testovania plánuje, monitoruje a riadi testovacie aktivity a úlohy ako bolo definované v sekcii 1.4.

Typické úlohy vedúceho testovania môžu zahŕňať:

- Koordinovanie testovacej stratégie a plánu s projektovými manažérmi a inými osobami.
- Vytváranie alebo revidovanie testovacej stratégie pre projekt a zásad testovania pre organizáciu.
- Podieľanie sa na projektových činnostiach z pohľadu testovania, ako napr. integračné plánovanie.
- Plánovanie testov – s ohľadom na kontext a chápanie ciele testovania a riziká – vrátane výberu prístupov k testovaniu, odhadovania času, úsilia a nákladov na testovanie, získavania zdrojov, definovania úrovni testovania, cyklov a plánovania manažmentu incidentov.
- Iniciovanie špecifikácie, prípravy, implementácie a vykonávania testov, monitorovanie výsledkov testovania a kontrola výstupných kritérií.
- Prispôsobovanie plánovania založené na výsledkoch a postupe testovania (niekedy dokumentovanom v status reporte) a prijímanie opatrení potrebných na odstránenie problémov.
- Nastavenie adekvátneho konfiguračného manažmentu testvéru za účelom sledovateľnosti.
- Zavedenie vhodných metrik pre meranie postupu testovania a hodnotenie kvality testovania a produktu.
- Rozhodovanie o tom, čo by malo byť automatizované, v akej miere a akým spôsobom.
- Výber nástrojov na podporu testovania a organizovanie školení v používaní nástrojov pre testerov.
- Rozhodovanie o implementácii testovacieho prostredia.
- Vytváranie sumárnych reportov založených na informáciách získaných počas testovania.

Typické úlohy testera môžu zahŕňať:

- Revidovanie a podieľanie sa na tvorbe testovacích plánov.
- Analýzu, revidovanie a posúdenie požiadaviek užívateľov, špecifikácií a modelov testovateľnosti.
- Tvorbu špecifikácií testov.
- Nastavovanie testovacieho prostredia (často koordinované s administráciou systému a sieťovým manažmentom).
- Prípravu a získavanie testovacích dát.
- Implementáciu testov na všetkých úrovniach testovania, vykonávanie a zaznamenávanie testov, vyhodnocovanie výsledkov testov a dokumentovanie odchýlok od očakávaných výsledkov.
- Používanie nástrojov pre administráciu a manažment testovania a nástrojov pre monitorovanie testovania podľa potreby.
- Automatizovanie testov (môže byť podporené vývojárom alebo expertom pre automatizáciu testov).
- Meranie výkonu komponentov alebo systémov (ak je to vhodné).
- Revidovanie testov vytvorených inými testerami.

Ľudia, ktorí pracujú na analýze testovania, návrhu testovania, špecifických typoch testovania alebo na automatizácii testovania môžu byť špecialistami v týchto rolách. V závislosti na úrovni testovania a rizikách vzťahujúcich sa k produktu a projektu, môžu rolu testera prevziať rôzni ľudia s cieľom udržať určitý stupeň nezávislosti. Obvykle by tester na úrovni komponentov a integračných testov mohli byť vývojári, tester na akceptačnej úrovni testovania biznis experti a užívatelia, tester na prevádzkovej akceptačnej úrovni testovania môžu byť operátori.

5.2 Plánovanie a odhadovanie testovania (Z3)

40 minút

Základné výrazy

Prístup k testovaniu, stratégia testovania.

5.2.1 Plánovanie testovania (Z2)

Táto časť vysvetľuje účel plánovania testovania v rámci vývojových a implementačných projektov a pre aktivity údržby. Plánovanie môže byť dokumentované v projektovom alebo hlavnom testovacom pláne a v oddelených plánoch pre rôzne úrovne testovania, ako sú systémové alebo akceptačné testovanie. Vzor dokumentu pre plánovanie testov pokrýva 'Štandard pre dokumentáciu testovania softvéru' (IEEE 829-1998).

Plánovanie je ovplyvnené pravidlami testovania organizácie, rozsahom testovania, cieľmi, rizikami, obmedzeniami, kritickosťou, testovateľnosťou a disponibilitou zdrojov. Čím pokročilejšie je projektové plánovanie a plánovanie testov, o to viac informácií začína byť k dispozícii a viacero detailov môže byť zahrnutých do plánu.

Plánovanie testovania je kontinuálna činnosť a je vykonávaná vo všetkých procesoch a aktivitách životného cyklu. Spätná väzba z testovacích činností sa používa na rozpoznanie meniacich sa rizík, aby sa plánovanie mohlo prispôbovať.

5.2.2 Aktivity plánovania testovania (Z3)

Aktivity plánovania testovania pre celý systém, alebo časť systému môžu zahŕňať:

- Určenie rozsahu a rizík a identifikovanie cieľov testovania.
- Definovanie celkového prístupu testovania, vrátane definície úrovni testovania, vstupných a výstupných kritérií.
- Integráciu a koordináciu testovacích aktivít do aktivít životného cyklu softvéru (akvizícia, ponuka, vývoj, prevádzka a údržba).
- Vykonávanie rozhodnutí o tom, čo testovať, ktoré roly budú vykonávať testovacie aktivity, ako budú testovacie aktivity vykonané a ako budú vyhodnocované výsledky testovania.
- Príprava harmonogramov aktivít analýzy a návrhu testovania.
- Príprava harmonogramov implementácie, vykonania a vyhodnotenia testov.
- Pridelovanie zdrojov na rôzne definované úlohy.
- Definovanie množstva, úrovne detailu, štruktúry a šablón pre testovaciu dokumentáciu.
- Výber metrik pre monitoring a riadenie prípravy a vykonávania testov, riešenia defektov a sledovania rizík.
- Nastavenie úrovne detailu pre testovacie procedúry s cieľom poskytnúť dostatok informácií na podporu reprodukovateľnej prípravy a vykonávania testov.

5.2.3 Vstupné kritériá (K2)

Vstupné kritériá definujú, kedy začať testovať, ako napríklad na začiatku úrovne testovania, alebo keď je sada testov pripravená pre vykonanie.

Typické vstupné kritériá môžu pokrývať nasledujúce:

- Dostupnosť a pripravenosť testovacieho prostredia.
- Pripravenosť testovacieho nástroja v testovacom prostredí.
- Dostupnosť testovateľného kódu.
- Dostupnosť testovacích dát.

5.2.4 Výstupné kritériá (Z2)

Výstupné kritériá definujú, kedy prestať testovať, ako napríklad na konci jednej úrovne testovania, alebo keď sada testov dosiahne špecifický cieľ.

Typické výstupné kritériá môžu pokrývať nasledujúce:

- Merania dôslednosti, ako napríklad pokrytie kódu, funkcionality alebo rizika.
- Odhady hustoty defektov alebo meraní spoľahlivosti.
- Náklady.
- Reziduálne riziká, ako napríklad neopravené defekty alebo nedostatočné pokrytie testmi v určitých oblastiach.
- Harmonogramy, napríklad založené na dobe uvedenia na trh.

5.2.5 Odhadovanie testovania (Z2)

Dva prístupy k odhadovaniu testovacieho úsilia sú:

- Prístup založený na metrikách: odhadovanie testovacieho úsilia na základe metrick minulých alebo podobných projektov alebo založené na typických hodnotách.
- Prístup založený na expertíze: odhadovanie úloh založené na odhadoch vykonaných ich vlastníkmi alebo expertmi.

Potom, ako sa odhadne testovacie úsilie, môžu byť identifikované zdroje a zostavený harmonogram.

Testovacie úsilie je závislé na množstve faktorov, medzi ktoré patria:

- Charakteristiky produktu: kvalita špecifikácie a iných informácií používaných v testovacích modeloch (t.j. základ testovania), veľkosť produktu, zložitosť domény problému, požiadavky na spoľahlivosť a bezpečnosť a požiadavky na dokumentáciu.
- Charakteristiky procesu vývoja: stabilita organizácie, používané nástroje, proces testovania, schopnosti ľudí a časový tlak.
- Výsledok testovania: počet defektov a množstvo požadovaného prepracovania.

5.2.6 Stratégia testovania, prístup k testovaniu (Z2)

Prístup k testovaniu je implementáciou stratégie testovania pre špecifický projekt. Prístup k testovaniu je definovaný a vypracovaný v testovacích plánoch a návrhoch testov. Obvykle zahŕňa učené rozhodnutia založené na ciele a zhodnotení rizika (testovacieho) projektu. Je východným bodom pre plánovanie procesu testovania, výber techník návrhu testov a typov testovania, ktoré budú použité a taktiež pre definovanie vstupných a výstupných kritérií.

Vybraný prístup závisí od kontextu a môže zvažovať riziká, hazardy a bezpečnosť, dostupné zdroje a zručnosti, technológiu, povahu systému (napr. softvér vyrobený na zakázku vs. krabicový softvér), ciele testovania a regulácie.

Typické prístupy zahŕňajú:

- Analytické prístupy, ako napríklad testovanie založené na rizikách, kde je testovanie nasmerované na najrizikovejšie oblasti.
- Prístupy založené na modeloch, ako sú stochastické testovanie používajúce štatistické informácie o mierach zlyhaní (ako modely rastu spoľahlivosti) alebo používaní (ako prevádzkové profily).
- Metodické prístupy, napr. založené na zlyhaniach (zahŕňajúce odhadovanie druhu chýb a útoky na chyby), založené na skúsenosti, kontrolnom zozname a kvalitatívnych vlastnostiach.
- Prístupy založené na procesoch alebo zodpovedajúce štandardom, napríklad štandardy špecifické pre priemysel alebo rôzne agilné metódy.
- Dynamické a heuristické prístupy, ako napríklad prieskumné testovanie, ktoré skôr reaguje na udalosti namiesto toho, aby bolo vopred plánované a kde vykonávanie a vyhodnocovanie sú súčasnými úlohami.

Certifikovaný tester

Učebná osnova pre základný stupeň



- Konzultatívne prístupy, napríklad také, v ktorých je pokrytie testami primárne riadené odporúčaniami a radami expertov z technologických a/alebo biznis kruhov mimo testovacieho tímu.
- Regresno-averzné prístupy, ktoré zahŕňajú opätovné použitie existujúceho testovacieho materiálu, rozsiahlu automatizáciu funkcionálnych regresných testov a štandardné testovacie sady.

Je možné kombinovať rôzne prístupy, napríklad dynamický prístup založený na rizikách.

5.3 Sledovanie a riadenie postupu testovania (Z2)

20 minút

Základné výrazy

Hustota defektov, miera zlyhaní, riadenie testovania, sledovanie testu, sumárny report z testovania.

5.3.1 Sledovanie postupu testovania (Z1)

Účelom sledovania testovania je poskytnúť spätnú väzbu a zviditeľniť testovacie činnosti. Monitorované informácie môžu byť zbierané manuálne alebo automaticky a môžu byť použité na meranie výstupných kritérií, ako napríklad pokrytie. Metriky môžu byť tiež použité na zhodnotenie postupu voči plánovanému harmonogramu a rozpočtu. Bežné testovacie metriky zahŕňajú:

- Percentuálny podiel vykonanej práce v príprave testovacích prípadov (alebo percentuálny podiel pripravených testovacích prípadov z plánovaných).
- Percentuálny podiel vykonanej práce v príprave testovacieho prostredia.
- Vykonávanie testovacích prípadov (napr. počet vykonaných/nevykonaných testovacích prípadov, a počet testovacích prípadov vykonaných úspešne/neúspešne).
- Informácie o defektoch (napr. hustota defektov, nájdené a opravené defekty, miera zlyhaní a výsledky retestov).
- Pokrytie testovania požiadaviek, rizík alebo kódu.
- Subjektívna dôvera testerov v daný produkt.
- Dátumy testovacích míľnikov.
- Náklady na testovanie vrátane nákladov v porovnaní voči prínosom nájdenia ďalšieho defektu alebo spustenia ďalšieho testu.

5.3.2 Reportovanie z testovania (Z2)

Reportovanie z testovania sa zaoberá sumárnymi informáciami o testovacích aktivitách. Zahŕňa:

- Čo sa udialo počas obdobia testovania, ako napríklad dátumy, kedy boli splnené výstupné kritériá.
- Analyzované informácie a metriky na podporu odporúčaní a rozhodnutí o následných krokoch, ako odhad zostávajúcich defektov, ekonomické prínosy pokračujúceho testovania, zostávajúce riziká, a úroveň dôvery v testovaný softvér.

Vzor sumárneho reportu z testovania sa nachádza v 'Štandarde pre dokumentáciu testovania softvéru' (IEEE Std 829-1998).

Metriky by mali byť zbierané počas úrovne testovania a na jej konci s cieľom zhodnotiť:

- Adekvátnosť cieľov testovania pre úroveň testovania.
- Adekvátnosť zvolených prístupov testovania.
- Efektívnosť testovania s ohľadom na ciele.

5.3.3 Riadenie testovania (Z2)

Riadenie testovania popisuje akékoľvek usmerňujúce alebo korekčné činnosti, ktoré boli vykonané na základe získaných a reportovaných informácií a metrík. Činnosti môžu pokrývať ktorékoľvek testovacie aktivity a môžu ovplyvniť akékoľvek aktivity alebo úlohy životného cyklu softvéru.

Príklady činností riadenia testovania zahŕňajú:

- Vykonávanie rozhodnutí založených na informáciách zo sledovania testov.
- Menenie priorít testov, keď sa vyskytne identifikované riziko (napr. neskorá dodávka softvéru).
- Menenie testovacieho harmonogramu s ohľadom na dostupnosť, alebo nedostupnosť testovacieho prostredia.
- Nastavovanie vstupného kritéria, ktoré požaduje retestovanie (konfirmačné testovanie) opráv vývojárom predtým, než budú akceptované do buildu.

5.4 Konfiguračný manažment (Z2)

10 minút

Základné výrazy

Konfiguračný manažment, riadenie verzií.

Pozadie

Účelom konfiguračného manažmentu je vytvorenie a udržiavanie integrity softvérových produktov (komponenty, dáta a dokumentácia) alebo systému počas projektového a produktového životného cyklu.

Pre testovanie môže konfiguračný manažment zaistiť nasledovné:

- Všetky položky testvéru sú identifikované, verzie sú riadené, zmeny sledované, vzájomné prepojenie a prepojenie s vývojovými položkami (objektmi testovania) existujúce tak, že je možné udržať sledovateľnosť počas celého procesu testovania.
- Na všetky identifikované dokumenty a položky softvéru sa dá jednoznačne odkázať v testovacej dokumentácii.

Testerom konfiguračný manažment pomáha jednoznačne identifikovať (a reprodukovať) testované položky, testovacie dokumenty, testy a testovací postroj(e).

Procedúry a infraštruktúra (nástroje) konfiguračného manažmentu by mali byť zvolené, dokumentované a implementované počas plánovania testov.

5.5 Riziko a testovanie (Z2)

30 minút

Základné výrazy

Produktové riziko, projektové riziko, riziko, testovanie založené na rizikách.

Pozadie

Riziko môže byť definované ako možnosť výskytu udalosti, hrozby, nebezpečenstva alebo situácie a ktoré vedie k nežiadúcim následkom, alebo ako potenciálny problém. Úroveň rizika sa určuje podľa pravdepodobnosti výskytu nepriaznivej udalosti a účinku (škoda vyplývajúca z tejto udalosti).

5.5.1 Projektové riziká (Z2)

Projektové riziká sú riziká ohraničujúce schopnosť projektu dosiahnuť jeho ciele, ako napríklad:

- Organizačné faktory:
 - nedostatok schopností, školení a personálu;
 - personálne problémy;
 - politické problémy, ako napríklad:
 - problémy s testerami komunikujúcimi ich potreby a výsledky testov;
 - zlyhanie tímu pri využití informácií vyplývajúcich z testovania a revízií (napr. nezlepšovanie vývojových a testovacích praktík).
 - nevhodný postoj k testovaniu alebo nevhodné očakávania od testovania (napr. nedocenená hodnota nájdenia defektov počas testovania).
- Technické problémy:
 - problémy v definovaní správnych požiadaviek;
 - rozsah, v ktorom nemôžu byť požiadavky naplnené pri existujúcich obmedzeniach;
 - nepripravenosť testovacieho prostredia včas;
 - neskorá konverzia dát, neskoré plánovanie migrácie a vývoja, ako aj neskorá konverzia testovacích dát / migračných nástrojov;
 - nízka kvalita návrhu, kódu, konfiguračných dát, testovacích dát a testov.
- Dodávateľské problémy:
 - zlyhanie tretej strany;
 - zmluvné problémy.

Keď sa tieto riziká analyzujú, riadia a zmierňujú, manažér testovania postupuje podľa dobre zavedených princípov projektového manažmentu. Vzor testovacieho plánu zo 'Štandardu pre dokumentáciu testovania softvéru' (IEEE Std 829-1998) požaduje stanoviť riziká a záložné plány.

5.5.2 Produktové riziká (Z2)

Oblasti potenciálnych zlyhaní (nepriaznivé budúce udalosti alebo nebezpečenstvá) v softvéri alebo systéme sú známe ako produktové riziká, keďže sú to riziká ohrozujúce kvalitu produktu. Tie zahŕňajú:

- Dodaný softvér náchylný k chybám.
- Možnosť, že softvér/hardvér poškodí jednotlivca alebo spoločnosť.
- Nedostatočné vlastnosti softvéru (napr. funkcionálna, spoľahlivosť, použiteľnosť a výkon).
- Slabá integrita a kvalita dát (napr. problémy migrácie, konverzie a prenosu dát, nedodržanie dátových štandardov).
- Softvér, ktorý nevykonáva určené funkcie.

Riziká sa využívajú na rozhodovanie, kde začať testovať a kde testovať viac; testovanie sa používa na zníženie rizika výskytu nepriaznivej udalosti alebo na zníženie dopadu nepriaznivej udalosti.

Produktové riziká sú špeciálnym typom rizika ohrozujúcim úspech projektu. Testovanie ako aktivita kontrolujúca riziko poskytuje spätnú väzbu o reziduálnom riziku meraním efektívnosti odstraňovania kritických defektov a rezervných plánov.

Prístup k testovaniu založený na rizikách poskytuje príležitosti proaktívne redukovať úroveň produktového rizika, začínajúc v úvodných štádiách projektu. Zahŕňa identifikáciu produktových rizík a ich použitie pri vedení plánovania a riadenia testovania, špecifikácii, príprave a vykonávaní testov. V prístupe založenom na rizikách môžu byť identifikované riziká použité na:

- Určenie testovacích techník, ktoré budú použité.
- Určenie rozsahu testovania, ktoré má byť vykonané.
- Stanovenie priorít testovania s cieľom nájsť kritické defekty, tak skoro ako je len možné.
- Určenie, či môžu byť na redukovanie rizika použité niektoré aktivity, ktoré nesúvisia priamo s testovaním (napr. poskytnutie školení pre neskúsených návrhárov).

Testovanie založené na rizikách vychádza z kolektívnych vedomostí a prehľadu kľúčových osôb na projekte s cieľom určiť riziká a úroveň testovania požadovanú na vysporiadanie sa s týmito rizikami.

Na zaistenie toho, že pravdepodobnosť zlyhania produktu je minimalizovaná, poskytujú aktivity rizikového manažmentu disciplinovaný prístup k:

- Zhodnoteniu (a pravidelnému opätovnému zhodnoteniu) toho, čo sa môže vyvíjať nepriaznivo (riziká).
- Určeniu, ktorými rizikami je dôležité zaoberať sa.
- Vykonaniu činností slúžiacich na vysporiadanie sa s rizikami.

Naviac, testovanie môže podporovať identifikáciu nových rizík, môže pomáhať určiť riziká, ktoré by mali byť redukované a môže znížiť neistotu vzhľadom na riziká.

5.6 Manažment incidentov (Z3)

40 minút

Základné výrazy

Zaznamenávanie incidentov, manažment incidentov, incident report.

Pozadie

Vzhľadom na to, že jedným z cieľov testovania je nájsť defekty, rozdiely medzi skutočnými a očakávanými výsledkami musia byť zaznamenané ako incidenty. Incident musí byť preskúmaný a môže sa ukázať, že sa jedná o defekt. Musia byť definované vhodné aktivity k vyriešeniu incidentov a defektov. Incidenty a defekty musia byť sledované od ich nájdania a klasifikácie, po opravu a potvrdenie riešenia. Za účelom riadiť všetky incidenty po ich riešení by organizácia mala zaviesť proces manažmentu incidentov a pravidiel pre ich klasifikáciu.

Incidenty môžu byť objavené počas vývoja, revidovania, testovania alebo používania softvérového produktu. Môžu vzniknúť z dôvodu problémov v kóde alebo prevádzkovanom systéme alebo v akomkoľvek type dokumentácie vrátane požiadaviek, vývojovej dokumentácie, dokumentov testovania a užívateľských informácií ako "Nápoveda" alebo inštaláčny manuál.

Záznamy o incidente majú nasledovné ciele:

- Poskytnúť vývojárom a iným stranám spätnú väzbu o probléme s cieľom umožniť identifikáciu, izoláciu a opravu podľa potreby.
- Poskytnúť vedúcim testovania prostriedok na sledovanie kvality testovaného systému a postupu testovania.
- Poskytnúť nápady pre zlepšenie procesu testovania.

Detaily záznamu o incidente môžu zahŕňať:

- Dátum vydania, organizáciu a autora záznamu.
- Očakávaný a skutočný výsledok.
- Identifikáciu predmetu testovania (konfiguračnej jednotky) a prostredia.
- Proces životného cyklu softvéru alebo systému, v ktorom bol incident spozorovaný.
- Popis incidentu za účelom reprodukcie a riešenia, môže obsahovať logy, dumpy databáz alebo screenshoty.
- Rozsah alebo stupeň dopadu na záujmy kľúčových osôb na projekte.
- Závažnosť dopadu na systém.
- Naliehavosť/priorita opravy.
- Stav incidentu (napr. otvorený, odložený, duplicitný, čakajúci na opravu, opravený a čakajúci na potvrdenie alebo uzavretý).
- Závery, odporúčania a schválenia.
- Globálne problémy, ako napríklad iné oblasti, ktoré môžu byť ovplyvnené zmenou vyvolanou incidentom.
- História zmien, ako napríklad sled akcií vykonaných členmi projektového tímu v súvislosti s incidentom s cieľom incident izolovať, opraviť a potvrdiť, že bol opravený.
- Odkazy, napríklad na špecifikáciu testovacieho prípadu, ktorý odhalil problém.

Štruktúra záznamu o incidente je taktiež zahrnutá v 'Štandarde pre dokumentáciu testovania softvéru' (IEEE Std 829-1998).

Referencie

- 5.1.1 Black, 2001, Hetzel, 1988
- 5.1.2 Black, 2001, Hetzel, 1988

Certifikovaný tester

Učebná osnova pre základný stupeň



- 5.2.5 Black, 2001, Craig, 2002, IEEE Std 829-1998, Kaner 2002
- 5.3.3 Black, 2001, Craig, 2002, Hetzel, 1988, IEEE Std 829-1998
- 5.4 Craig, 2002
- 5.5.2 Black, 2001, IEEE Std 829-1998
- 5.6 Black, 2001, IEEE Std 829-1998

6. Podporné nástroje pre testovanie (Z2)

80 minút

Študijné ciele pre podporné nástroje pre testovanie

Tieto ciele určujú, čo budete vedieť po dokončení každého modulu.

6.1 Typy testovacích nástrojov (Z2)

ŠC-6.1.1 Klasifikovať rôzne typy testovacích nástrojov podľa ich účelu, aktivít základného testovacieho procesu a životného cyklu softvéru. (Z2)

ŠC-6.1.3 Vysvetliť pojem testovací nástroj a účel podpory testovania nástrojmi. (Z2) ²

6.2 Efektívne používanie nástrojov: možné výhody a riziká (Z2)

ŠC-6.2.1 Zhrnúť možné výhody a riziká automatizácie testov a podporných nástrojov pre testovanie. (Z2)

ŠC-6.2.2 Zapamätať si špecifiká nástrojov pre vykonávanie testov, statickej analýzy a nástrojov pre manažment testovania. (Z1)

6.3 Zavedenie nástroja v organizácii (Z1)

ŠC-6.3.1 Uviesť hlavné princípy zavedenia nástroja v organizácii. (Z1)

ŠC-6.3.2 Určiť ciele skúšky konceptu (proof of concept) pre vyhodnotenie nástroja a pilotnej fázy pre implementáciu nástroja. (Z1)

ŠC-6.3.3 Pochopiť, že pre zabezpečenie dostatočnej podpory nástroja nepostačuje iba jeho obstaranie, ale sú potrebné aj iné činitele. (Z1)

² ŠC-6.1.3 je zámerne preskočený

6.1 Typy testovacích nástrojov (Z2)**45 minút****Základné výrazy**

Nástroj pre konfiguračný manažment, nástroj pre meranie pokrytia, nástroj pre ladenie, nástroj pre dynamickú analýzu, nástroj pre manažment incidentov, nástroj pre záťažové testovanie, modelovací nástroj, monitorovací nástroj, nástroj pre testovanie výkonu, vplyv skúšania, nástroj pre manažment požiadaviek, nástroj pre revidovanie, nástroj pre bezpečnosť, nástroj pre statickú analýzu, nástroj pre stres testovanie, komparátor testu, nástroj na prípravu testovacích dát, nástroj pre navrhovanie testov, testovací postroj, nástroj na vykonanie testu, nástroj pre manažment testovania, nástroj pre framework jednotkového testovania.

6.1.1 Podporné nástroje pre testovanie

Testovacie nástroje môžu byť použité pre jednu alebo viacero aktivít, ktoré podporujú testovanie. Tieto aktivity zahŕňajú:

1. Nástroje, ktoré sú priamo použité v testovaní ako sú nástroje pre vykonanie testov, nástroje pre generovanie testovacích dát a nástroje pre porovnávanie výsledkov.
2. Nástroje, ktoré pomáhajú v riadení procesu testovania ako sú nástroje použité pre riadenie testov, výsledkov testov, dát, požiadaviek, incidentov, defektov atď. a pre reportovanie a monitorovanie vykonania testov.
3. Nástroje, ktoré sú použité v rekognoskácii, alebo jednoducho povedané: preskúmanie (napr. nástroje, ktoré monitorujú aktivitu súboru sledovanej aplikácie).
4. Akýkoľvek nástroj, ktorý napomáha v testovaní (tabuľkový procesor je taktiež testovacím nástrojom v tomto význame).

Podporný nástroj pre testovanie môže mať jeden alebo viacero nasledujúcich účelov, v závislosti na kontexte:

- Zlepšenie efektívnosti testovacích aktivít automatizáciou opakovaných úloh alebo podporovaním manuálnych testovacích aktivít ako napr. plánovanie testovania, návrh testu, reportovanie a monitorovanie testovania.
- Automatizovanie aktivít, ktoré vyžadujú významné množstvo zdrojov v prípade, že sú vykonávané manuálne (napr. statické testovanie).
- Automatizovanie aktivít, ktoré nemôžu byť vykonávané manuálne (napr. testovanie výkonu veľkého rozsahu klient-server aplikácií).
- Zvýšenie spoľahlivosti testovania (napr. automatizáciou porovnávania veľkého množstva dát alebo simulovaním správania).

Pojem „framework testovania“ sa taktiež často používa v priemysle, v najmenej troch významoch:

- Znovupoužiteľné a rozšíriteľné testovacie knižnice, ktoré môžu byť použité na vytvorenie nástrojov pre testovanie (taktiež nazývané testovacie postroje).
- Typ návrhu automatizovania testu (napr. riadený dátami, riadený kľúčovými slovami).
- Celkový proces vykonania testovania.

Pre účely tejto učebnej osnovy je pojem „framework testovania“ použitý v jeho prvých dvoch významoch, tak ako je popísané v sekcii 6.1.6.

6.1.2 Klasifikácia testovacích nástrojov (Z2)

Existuje viacero nástrojov, ktoré podporujú rôzne stránky testovania. Nástroje môžu byť klasifikované na základe viacerých kritérií ako sú účel, komerčný / voľne dostupný / open-source / shareware, použitá technológia a tak ďalej. Nástroje sú v tejto osnove roztriedené podľa testovacích aktivít, ktoré podporujú.

Niektoré nástroje jasne podporujú jednu aktivitu. Iné môžu podporovať viac ako jednu aktivitu. Tieto sú klasifikované v rámci aktivity, s ktorou sú najviac spojené. Nástroje od jedného poskytovateľa, obzvlášť tie, ktoré boli navrhnuté, aby pracovali spolu, môžu byť súčasťou jedného balíka.

Niektoré typy testovacích nástrojov môžu byť rušivé, čo znamená, že môžu ovplyvniť skutočný výsledok testu. Napríklad nameraný čas môže byť rôzny z dôvodu, že sú nástrojom navyše vykonávané inštrukcie, alebo je možné získať rozdielnu mieru pokrytia kódu. Následok rušivých nástrojov nazývame vplyv skúšania.

Niektoré nástroje ponúkajú podporu vhodnú skôr pre vývojárov (napr. nástroje, ktoré sú použité počas testovania komponentov alebo integračného testovania komponentov). Takéto nástroje sú v rozdelení uvedenom ďalej označené písmenom "V".

6.1.3 Podporné nástroje pre manažment testovania a testov (Z1)

Nástroje pre manažment sa používajú na všetky testovacie aktivity počas celého životného cyklu softvéru.

Nástroje pre manažment testovania

Tieto nástroje poskytujú rozhrania pre vykonávanie testov, sledovanie defektov a riadenie požiadaviek spolu s podporou pre kvantitatívnu analýzu a reportovanie objektov testovania. Tiež podporujú sledovanie týchto objektov testovania k špecifikáciám požiadaviek a môžu mať schopnosť nezávislého riadenia verzií, alebo rozhranie na externý nástroj.

Nástroje pre manažment požiadaviek

Tieto nástroje uchovávajú popisy požiadaviek, uchovávajú atribúty pre požiadavky (vrátane priority), poskytujú jedinečné identifikátory a podporujú sledovanie týchto požiadaviek k jednotlivým testom. Tieto nástroje môžu taktiež pomôcť identifikovať nekonzistentné alebo chýbajúce požiadavky.

Nástroje pre manažment incidentov (Nástroje pre sledovanie defektov)

Tieto nástroje uchovávajú a riadia záznamy o incidentoch, t.j. defektoch, zlyhaniach, požiadavkách na zmenu alebo spozorovaných problémoch a anomáliách a pomáhajú v riadení životného cyklu incidentov, prípadne s podporou pre štatistickú analýzu.

Nástroje pre konfiguračný manažment

Aj keď tieto nástroje nie sú vyslovene nástrojmi pre testovanie, sú nevyhnutné pre uchovávanie a manažment verzií testvéru a súvisiaceho softvéru, obzvlášť v prípade že je konfigurované viac ako jedno hardvérové/softvérové prostredie vzhľadom na verzie operačného systému, kompilátory, prehliadače atď.

6.1.4 Podporné nástroje pre statické testovanie (Z1)

Nástroje pre statické testovanie poskytujú nákladovo efektívny spôsob nachádzania väčšieho počtu defektov v skoršom štádiu procesu vývoja.

Nástroje pre revidovanie

Tieto nástroje pomáhajú s procesmi revízie, kontrolnými zoznamami, revíznymi smernicami a sú používané pre uchovávanie a komunikovanie pripomienok revidovania, reportujú defekty a potrebné úsilie. Môžu byť ďalej nápomocné pri poskytovaní pomoci pri "online" revidovaní pre veľké alebo geograficky rozptýlené tímy.

Nástroje pre statickú analýzu (V)

Tieto nástroje pomáhajú vývojárom a testerom nájsť defekty pred dynamickým testovaním poskytovaním podpory pre presadzovanie štandardov kódovania (zahŕňajúce bezpečné kódovanie),

analýzu štruktúr a závislostí. Môžu tiež pomôcť v plánovaní alebo analýze rizík poskytovaním metrik kódu (napr. zložitosť).

Modelovacie nástroje (V)

Tieto nástroje sú používané pre validáciu modelov softvéru (napr. fyzický dátový model (PDM) pre relačnú databázu) tak, že vymenujú nekonzistencie a identifikujú defekty. Tieto nástroje môžu často pomôcť pri generovaní niektorých testovacích prípadov založených na modeloch.

6.1.5 Podporné nástroje pre špecifikáciu testov (Z1)

Nástroje pre navrhovanie testov

Tieto nástroje sú používané na generovanie testovacích vstupov alebo spustiteľných testov a/alebo testovacích prognóz z požiadaviek, z užívateľských grafických rozhraní, z návrhových modelov (stavový, dátový, objektový) alebo z kódu.

Nástroje pre prípravu testovacích dát

Nástroje pre prípravu testovacích dát manipulujú s databázami, súbormi alebo prenosmi dát tak, aby pripravili testovacie dáta na použitie počas vykonávania testov za účelom zabezpečenia bezpečnosti prostredníctvom anonymity dát.

6.1.6 Podporné nástroje pre vykonanie a zaznamenávanie testov (Z1)

Nástroje na vykonanie testov

Tieto nástroje umožňujú automatické alebo polo-automatické vykonanie, použitím uložených vstupov a očakávaných výstupov s využitím skriptovacieho jazyka a zvyčajne poskytujú protokol o teste pre každý testovací beh. Môžu byť tiež použité na nahrávanie testov a zvyčajne podporujú skriptovacie jazyky alebo konfiguráciu založenú na GUI pre parametrizáciu dát a možnosť ďalších úprav v testoch.

Testovací postroj / nástroj pre framework jednotkového testovania (V)

Testovací postroj alebo framework napomáha testovaniu komponentov alebo častí systému simulovaním prostredia, v ktorom bude bežať objekt testovania, prostredníctvom poskytnutia simulovaných objektov ako sú nástavce alebo ovládače.

Komparátory testov

Komparátory testov určujú rozdiely medzi súbormi, databázami alebo výsledkami testovania. Nástroje pre vykonávanie testov zvyčajne zahŕňajú dynamické komparátory, ale porovnávanie po vykonaní môže byť uskutočnené v nezávislom porovnávacom nástroji. Komparátor testov môže využívať testovaciu prognózu, najmä v prípade, ak je automatizovaný.

Nástroje na meranie pokrytia (V)

Tieto nástroje prostredníctvom rušivých alebo nerušivých spôsobov, merajú percento preskúšaných špecifických typov štruktúr kódu (napr. príkazov, vetiev alebo rozhodovaní a modulov alebo volaní funkcií) sadou testov.

Nástroje pre bezpečnosť

Tieto nástroje sú použité pre vyhodnocovanie bezpečnostných charakteristík softvéru. To zahŕňa vyhodnocovanie schopnosti softvéru ochrániť dôvernosť dát, integritu, autentifikáciu, autorizáciu, dostupnosť a nepopierateľnosť. Nástroje pre bezpečnosť sú väčšinou zamerané na príslušnú technológiu, platformu alebo účel.

6.1.7 Podporné nástroje pre výkon a monitorovanie (Z1)

Nástroje pre dynamickú analýzu (V)

Nástroje pre dynamickú analýzu nachádzajú defekty, ktoré sú zjavné, až keď je softvér používaný – napríklad časové závislosti alebo pretečenia pamäte. Sú zvyčajne používané v testovaní komponentov, integračnom testovaní komponentov a pri testovaní strednej vrstvy (middleware).

Nástroje pre testovanie výkonu/zátťažové testovanie/stres testovanie

Nástroje pre testovanie výkonu sledujú a oznamujú správanie systému v rôznych simulovaných podmienkach používania v zmysle počtu súbežne pracujúcich užívateľov, modelu ich narastania, frekvencie a relatívneho percenta transakcií. Simulácia záťaže je dosiahnutá spôsobom vytvárania virtuálnych užívateľov, ktorí vykonávajú vybranú sadu transakcií rozloženú naprieč rôznymi testovacími automatmi. Tieto automaty sa zvyčajne nazývajú generátory záťaže.

Monitorovacie nástroje

Monitorovacie nástroje nepretržite analyzujú, verifikujú a reportujú používanie špecifických systémových zdrojov a varujú pred možnými problémami so službami.

6.1.8 Podporé nástroje pre špecifické potreby testovania (Z1)

Hodnotenie kvality dát

Dáta sú v centre pozornosti niektorých projektov, ako sú projekty dátovej konverzie/migrácie a aplikácií ako sú dátové sklady. Ich atribúty sa môžu líšiť v zmysle kritičnosti a objemu. V takomto kontexte je potrebné nasadiť nástroje pre hodnotenie kvality dát za účelom revízie a verifikácie konverzie dát a migračných pravidiel, tak aby bolo zaistené, že spracované dáta sú správne, kompletne a že sú v súlade s preddefinovaným štandardom ktorý je špecifický pre daný kontext.

Pre testovanie použiteľnosti existujú iné testovacie nástroje.

6.2 *Efektívne použitie nástrojov: možné výhody a riziká (Z2)*

20 minút

Základné výrazy

Testovanie riadené dátami, testovanie riadené kľúčovými slovami, skriptovací jazyk.

6.2.1 Možné výhody a riziká nástroja pre podporu testovania (pre všetky nástroje) (Z2)

Jednoduchý nákup alebo prenájom nástroja úspech s daným nástrojom negarantuje. Každý typ nástroja môže vyžadovať dodatočné úsilie na dosiahnutie skutočných a dlhotrvajúcich výhod. S použitím nástroja prichádzajú určité potenciálne výhody a príležitosti, ale sú tu aj riziká.

Možné výhody používania nástrojov zahŕňajú:

- Opakované činnosti sú redukované (napr. behy regresných testov, opätovné zadávanie rovnakých testovacích dát, kontrola štandardov kódovania).
- Väčšia konzistencia a opakovateľnosť (napr. testy vykonávané nástrojom v tom istom poradí s tou istou frekvenciou a testy odvodené z požiadaviek).
- Objektívne hodnotenie (napr. statické merania, pokrytie).
- Jednoduchý prístup k informáciám o testoch alebo testovaní (štatistiky a grafy postupu testovania, pomery incidentov a výkon).

Riziká používania nástrojov zahŕňajú:

- Nerealistické očakávania od nástroja (vrátane funkcionality a jednoduchosti používania).
- Podcenenie času, nákladov a úsilia pri prvotnom zavádzaní nástroja (vrátane tréningov a externej podpory).
- Podcenenie času a úsilia, ktoré sú potrebné na dosiahnutie značných a trvalých výhod z nástroja (vrátane potreby zmeny procesov testovania a kontinuálneho zlepšovania spôsobov použitia nástroja).
- Podcenenie úsilia vyžadovaného na údržbu testovacích produktov generovaných nástrojom.
- Prílišné spoliehanie sa na nástroj (náhrada návrhu testu alebo v prípade, ak by manuálne testovanie bolo vhodnejšie).
- Zanedbanie kontroly verzií artefaktov testovania v nástroji.
- Zanedbanie vzťahov a spolupôsobenia problémov medzi kritickými nástrojmi, ako sú nástroje pre manažment požiadaviek, nástroje pre kontrolu verzií, nástroje pre manažment incidentov, nástroje pre sledovanie defektov a nástroje od viacerých predajcov.
- Riziko dodávateľa nástroja, že skončí krachom, ukončí podporu nástroja, alebo predá nástroj inému dodávateľovi.
- Slabá odozva dodávateľa týkajúca sa podpory, upgradov a opráv defektov.
- Riziko ukončenia nástroja u open-source / free projektov.
- Neočakávané, ako napríklad neschopnosť podporovať novú platformu.

6.2.2 Osobité úvahy ku niektorým typom nástrojov (Z1)

Nástroje pre vykonanie testu

Nástroje pre vykonanie testu vykonávajú testované objekty s použitím automatizovaných testovacích skriptov. Tento typ nástroja často vyžaduje značné úsilie pre dosiahnutie podstatných výhod.

Zaznamenávanie testov nahrávaním akcií manuálneho testera vyzerá atraktívne, ale tento prístup prestáva byť efektívny pri veľkom množstve automatizovaných testovacích skriptov. Zaznamenaný

skript je priamočiarou reprezentáciou so špecifickými dátami a akciami, ktoré sú súčasťou každého skriptu. Tento typ skriptu môže byť nestabilný v prípade výskytu neočakávaných udalostí.

Prístup testovania riadeného dátami oddeľuje testovacie vstupy (dáta) zvyčajne do tabuľkového procesora a používa všeobecnejší testovací skript, ktorý vie čítať vstupné dáta a vykonať rovnaký skript s rôznymi dátami. Tester, ktorí nie sú oboznámení so skriptovacím jazykom, môžu potom vytvárať dáta pre tieto preddefinované skripty.

Existujú ďalšie techniky použité v technikách riadených dátami, kde namiesto kombinácie napevno nakódovaných dát umiestnaných v tabuľkovom procesore, sú dáta generované použitím algoritmov založených na konfigurovateľných parametroch v čase behu a sú dodané aplikácii. Napríklad nástroj môže používať algoritmus, ktorý generuje náhodné užívateľské ID a pre opakovanosť vzoru je pre kontrolu náhodnosti použitý rozsev.

V prípade prístupu testovania riadeného kľúčovými slovami tabuľkový procesor obsahuje kľúčové slová popisujúce akcie, ktoré majú byť vykonané (taktiež nazývané "akčné slová") a testovacie dáta. Tester (aj keď nie sú oboznámení so skriptovacím jazykom), môžu definovať testy použitím kľúčových slov, ktoré môžu byť prispôsobené na mieru testovanej aplikácii.

Technická znalosť skriptovacích jazykov je potrebná pri všetkých prístupoch (buď u testerov alebo u špecialistov na automatizáciu testovania).

Bez ohľadu na použitú techniku skriptovania, musia byť kvôli neskoršiemu porovnaniu uložené očakávané výsledky pre každý test.

Nástroje pre statickú analýzu

Nástroje pre statickú analýzu nasadené na zdrojový kód môžu presadzovať štandardy kódovania, ale v prípade aplikovania na existujúci kód môžu generovať veľké množstvo správ. Varovné správy nezastavia preklad kódu do vykonateľného programu, ale v ideálnom prípade by nemali zostať nepovšimnuté, pretože to uľahčí údržbu kódu v budúcnosti. Efektívnym prístupom je postupná implementácia nástroja pre analýzu s úvodnou filtráciou, ktorá vylučuje niektoré typy správ.

Nástroje pre manažment testovania

Nástroje pre manažment testovania vyžadujú rozhranie na ostatné nástroje alebo tabuľkové procesory za účelom produkovať užitočné informácie vo formáte, ktorý je vhodný pre potreby organizácie.

6.3 Zavedenie nástroja v organizácii (Z1)

15 minút

Základné výrazy

Žiadne špecifické výrazy.

Pozadie

Medzi najdôležitejšie úvahy pri výbere nástroja pre organizáciu patrí:

- Posúdenie zrelosti organizácie, silných a slabých stránok spolu s identifikáciou možností na zlepšenie testovacieho procesu s podporou nástrojov.
- Vyhodnotenie voči jasným požiadavkám a objektívnym kritériám.
- Skúška konceptu (proof-of-concept), s použitím nástroja na testovanie počas vyhodnocovacej fázy, ktorá určí, či je vykonávanie s testovaným softvérom a v súčasnej infraštruktúre efektívne, alebo identifikuje zmeny, ktoré táto infraštruktúra potrebuje k efektívnemu používaniu nástroja.
- Vyhodnotenie dodávateľa (vrátane školení, podpory a komerčných stránok) alebo poskytovateľov podporných služieb v prípade nekomerčných nástrojov.
- Identifikácia interných požiadaviek pre koučing a mentoring pri používaní nástroja.
- Vyhodnotenie školení potrebných s ohľadom na súčasné skúsenosti testovacieho tímu s automatizáciou testov.
- Vyhodnotenie pomeru nákladov a výnosov, ktorý je založený na konkrétnom obchodnom prípade.

Zavedenie zvoleného nástroja do organizácie začína pilotným projektom, ktorý má nasledujúce ciele:

- Dozvedieť sa viac podrobností o nástroji.
- Zhodnotiť, ako bude nástroj zapadať do existujúcich procesov a praktík a určiť čo bude potrebné zmeniť.
- Rozhodnúť o štandardnom spôsobe používania, spravovania, zálohovania a údržby nástroja a testovacích produktov (t.j. dohoda o menšej konvencii pre súbory a testy, vytváranie knižníc a definícia modularity testovacích sád).
- Posúdiť, či budú výhody dosiahnuté za primerané náklady.

Faktory úspechu pre zavedenie nástroja v rámci organizácie zahŕňajú:

- Postupné zavádzanie nástroja do zvyšku organizácie.
- Adaptácia a zlepšovanie procesov tak, aby zodpovedali použitiu nástroja.
- Zabezpečovanie tréningov a koučovania/mentorovania pre nových užívateľov.
- Definícia pravidiel používania.
- Implementovanie spôsobu získavania informácií o používaní nástroja z jeho aktuálneho používania.
- Monitorovanie používania nástroja a jeho výhod.
- Poskytovanie podpory testovaciemu tímu pre daný nástroj.
- Získavanie získaných ponaučení zo všetkých tímov.

Referencie

6.2.2 Buwalda, 2001, Fewster, 1999

6.3 Fewster, 1999

7. Referencie

Štandardy

ISTQB Glosár pojmov používaných v testovaní softvéru, verzia 2.1

[CMMI] Chrissis, M.B., Konrad, M. and Shrum, S. (2004) *CMMI, Guidelines for Process Integration and Product Improvement*, Addison Wesley: Reading, MA

Pozri kapitolu 2.1

[IEEE Std 829-1998] IEEE Std 829™ (1998) IEEE Standard for Software Test Documentation

Pozri kapitoly 2.3, 2.4, 4.1, 5.2, 5.3, 5.5, 5.6

[IEEE 1028] IEEE Std 1028™ (2008) IEEE Standard for Software Reviews and Audits

Pozri kapitolu 3.2

[IEEE 12207] IEEE 12207/ISO/IEC 12207-2008, Software life cycle processes

Pozri kapitolu 2.1

[ISO 9126] ISO/IEC 9126-1:2001, Software Engineering – Software Product Quality

Pozri kapitolu 2.3

Literatúra

[Beizer, 1990] Beizer, B. (1990) *Software Testing Techniques* (2nd edition), Van Nostrand Reinhold: Boston

Pozri kapitoly 1.2, 1.3, 2.3, 4.2, 4.3, 4.4, 4.6

[Black, 2001] Black, R. (2001) *Managing the Testing Process* (3rd edition), John Wiley & Sons: New York

Pozri kapitoly 1.1, 1.2, 1.4, 1.5, 2.3, 2.4, 5.1, 5.2, 5.3, 5.5, 5.6

[Buwalda, 2001] Buwalda, H. et al. (2001) *Integrated Test Design and Automation*, Addison Wesley: Reading, MA

Pozri kapitolu 6.2

[Copeland, 2004] Copeland, L. (2004) *A Practitioner's Guide to Software Test Design*, Artech House: Norwood, MA

Pozri kapitoly 2.2, 2.3, 4.2, 4.3, 4.4, 4.6

[Craig, 2002] Craig, Rick D. and Jaskiel, Stefan P. (2002) *Systematic Software Testing*, Artech House: Norwood, MA

Pozri kapitoly 1.4.5, 2.1.3, 2.4, 4.1, 5.2.5, 5.3, 5.4

[Fewster, 1999] Fewster, M. and Graham, D. (1999) *Software Test Automation*, Addison Wesley: Reading, MA

Pozri kapitoly 6.2, 6.3

[Gilb, 1993]: Gilb, Tom and Graham, Dorothy (1993) *Software Inspection*, Addison Wesley: Reading, MA

Pozri kapitoly 3.2.2, 3.2.4

[Hetzel, 1988] Hetzel, W. (1988) *Complete Guide to Software Testing*, QED: Wellesley, MA

Pozri kapitoly 1.3, 1.4, 1.5, 2.1, 2.2, 2.3, 2.4, 4.1, 5.1, 5.3

Certifikovaný tester

Učebná osnova pre základný stupeň



[Kaner, 2002] Kaner, C., Bach, J. and Pettitcord, B. (2002) Lessons Learned in Software Testing, John Wiley & Sons: New York
Pozri kapitoly 1.1, 4.5, 5.2

[Myers 1979] Myers, Glenford J. (1979) The Art of Software Testing, John Wiley & Sons: New York
Pozri kapitoly 1.2, 1.3, 2.2, 4.3

[van Veenendaal, 2004] van Veenendaal, E. (ed.) (2004) The Testing Practitioner (Chapters 6, 8, 10), UTN Publishers: The Netherlands
Pozri kapitoly 3.2, 3.3

8. Príloha A – Pozadie učebnej osnovy

História dokumentu

Tento dokument bol pripravený počas rokov 2004–2007 pracovnou skupinou skladajúcou sa z menovaných členov Medzinárodného výboru pre kvalifikáciu softvérového testovania (International Software Testing Qualifications Board - ISTQB®). Najskôr bol dokument revidovaný výborom pre revíziu a následne predstaviteľmi vybranými z medzinárodnej komunity testovania softvéru. Pravidlá použité pri tvorbe tohto dokumentu sú uvedené v prílohe C.

Tento dokument je učebnou osnovou pre medzinárodný Základný certifikát v testovaní softvéru, ktorý je prvou úrovňou medzinárodnej kvalifikácie odsúhlasenou ISTQB (www.istqb.org).

Ciele kvalifikácie Základný certifikát

- Získať uznanie testovania ako základnej a profesionálnej špecializácie softvérového inžinierstva.
- Poskytnúť štandardný framework pre kariérny rozvoj testerov.
- Umožniť profesionálne kvalifikovaným testerom, aby boli rešpektovaní zamestnávateľmi, zákazníkmi a kolegami a zároveň zlepšiť postavenie testerov.
- Presadzovať jednotné a vhodné testovacie postupy v rámci všetkých disciplín softvérového inžinierstva.
- Identifikovať oblasti testovania, ktoré sú podstatné a prínosné pre dané odvetvia.
- Umožniť dodávateľom softvéru, aby prijímali certifikovaných testerov a tak získali obchodnú výhodu voči svojej konkurencii zverejnením svojich pravidiel naboru testerov.
- Poskytnúť možnosť pre testerov a ďalších, ktorí majú záujem o testovanie, aby získali medzinárodne uznávanú kvalifikáciu v predmete testovania.

Ciele medzinárodnej kvalifikácie (prevzaté z ISTQB stretnutia v Sollentuna, November 2001)

- Umožniť porovnanie znalostí testovania v rôznych krajinách.
- Umožniť testerom jednoduchšie uplatnenie v iných krajinách.
- Umožniť nadnárodným/medzinárodným projektom, aby mali jednotné chápanie problematiky testovania.
- Celosvetovo zvyšovať množstvo kvalifikovaných testerov.
- Ako medzinárodne založená iniciatíva mať vyšší vplyv a hodnotu ako lokálny prístup v rámci jedného regiónu/krajiny.
- Rozvíjať spoločný medzinárodný súbor poznatkov a vedomostí o testovaní pomocou učebnej osnovy a terminológie a zvyšovať úroveň znalostí o testovaní u všetkých zúčastnených.
- Propagovať testovanie ako profesii v ďalších krajinách.
- Umožniť testerom, aby získali uznávanú kvalifikáciu v ich rodnom jazyku.
- Umožniť zdieľanie znalostí a zdrojov medzi krajinami.
- Poskytovať medzinárodné uznanie testerov a ich kvalifikácie spoluúčasťou množstva krajín.

Vstupné požiadavky na túto kvalifikáciu

Vstupné kritérium na získanie Základného certifikátu ISTQB® v testovaní softvéru je záujem kandidátov o testovanie softvéru. Avšak je silne odporúčané, aby kandidáti taktiež:

- Mali prinajmenšom minimálne znalosti z vývoja softvéru alebo testovania softvéru, napríklad šesťmesačnú skúsenosť ako tester systémových alebo akceptačných testov, prípadne ako softvérový vývojár.
- Absolvovali školenie, ktoré je akreditované podľa ISTQB® štandardov (jedným z uznaných lokálnych výborov ISTQB).

Pozadie a história Základného certifikátu v testovaní softvéru

Nezávislá certifikácia softvérových testerov začala vo Veľkej Británii v Skúšobnom výbore informačných systémov (Information Systems Examination Board (ISEB)) Britskej počítačovej

spoločnosti (British Computer Society) založením Výboru pre testovanie softvéru (Software testing board) v roku 1998 (www.bcs.org.uk/iseb). V roku 2002 začala spoločnosť ASQF v Nemecku podporovať Nemecký program pre kvalifikáciu testerov (German tester qualification scheme www.asqf.de). Táto učebná osnova je založená na osnovách ISEB a ASQF; zahŕňa reorganizovaný, aktualizovaný obsah a ďalší obsah a dôraz je kladený na témy, ktoré poskytnú najlepšiu praktickú pomoc testerom.

Už existujúci Základný certifikát v testovaní softvéru (napr. od ISEB, ASQF alebo lokálnych výborov uznaných ISTQB), ktorý bol udelený pred vydaním tohoto medzinárodného certifikátu, bude považovaný za ekvivalent medzinárodného certifikátu. Základný certifikát neexpiruje a nebude potrebné, aby bol predĺžovaný alebo obnovovaný. Dátum vydania sa nachádza na certifikáte.

V rámci každej zúčastnenej krajiny sú miestne aspekty kontrolované lokálnym výborom softvérového testovania uznaným ISTQB. Povinnosti lokálnych výborov sú zadané od ISTQB, ale sú implementované v rámci každej krajiny. V povinnostiach lokálnych výborov sa predpokladá zahrnutie akreditácie poskytovateľov školení a nastavenie skúšok.

9. Príloha B – Študijné ciele / kognitívna úroveň znalostí

Pre tieto osnovy sú použité nasledujúce študijné ciele. Každá téma v učebnej osnove bude preskúšaná podľa jej zodpovedajúceho študijného cieľa.

Úroveň 1: Zapamätať si (Z1)

Kandidát pochopí, zapamätá si a spomenie si na výraz, pojem alebo koncept.

Kľúčové slová: Zapamätať si, obnoviť, spomenúť si, rozoznať, vedieť.

Príklad

Pozná definíciu zlyhania ako:

- “Nedodanie služby koncovému užívateľovi alebo inej kľúčovej osobe”, alebo
- “Odchýlka komponentu alebo systému od jeho očakávanej dodávky, služby alebo výsledku”.

Úroveň 2: Pochopiť (Z2)

Kandidát vie označiť príčiny alebo vysvetlenia pre výroky týkajúce sa témy a vie zhrnúť, porovnať, klasifikovať, rozdeliť a uviesť príklady pre koncepty testovania.

Kľúčové slová: Zhrnúť, zovšeobecniť, abstrahovať, klasifikovať, porovnať, namapovať, odlíšiť, doložiť príkladom, vysvetliť, preložiť, znázorniť, odvodiť, vyvodiť, kategorizovať, vytvoriť model.

Príklady

Vie vysvetliť dôvod, prečo by mali byť testy vytvorené tak skoro, ako je to možné:

- Nájsť defekty keď sú lacnejšie odstrániteľné.
- Nájsť najdôležitejšie defekty ako prvé.

Dokáže vysvetliť podobnosti a rozdiely medzi integračným a systémovým testovaním:

- Podobnosti: testovanie viac ako jedného komponentu, môžu byť testované nefunkcionálne aspekty.
- Rozdiely: integračné testovanie sa zameriava na rozhrania a interakcie a systémové testovanie sa zameriava na aspekty celého systému, ako je proces od začiatku do konca (end-to-end).

Úroveň 3: Použiť (Z3)

Kandidát vie vybrať správnu aplikáciu konceptu alebo techniky a aplikovať ich v danom kontexte.

Kľúčové slová: Zavádzať, vykonať, používať, sledovať postup, uplatniť postup.

Príklad

- Vie identifikovať hraničné hodnoty pre platné a neplatné sekcie.
- Vie vybrať testovacie prípady pre daný diagram prechodu stavov s pokrytím všetkých prechodov.

Úroveň 4: Analyzovať (Z4)

Kandidát vie rozdeliť informácie vzťahujúce sa k procedúre, alebo technike na jej základné časti za účelom lepšieho pochopenia. Dokáže taktiež rozlíšiť fakty od logických záverov. Typickým uplatnením je analýza dokumentu, softvérovej alebo projektovej situácie a návrh vhodných krokov riešenia problému alebo úlohy.

Kľúčové slová: Analyzovať, organizovať, nájsť súvislosti, integrovať, naznačiť, rozložiť, štrukturovať, prisúdiť, dekonštruovať, rozlíšiť, potlačiť, rozoznať, zamerať, vybrať.

Príklad

- Analyzujte produktové riziká a navrhňte preventívne a nápravné aktivity za účelom zníženia rizík.

- Popíšte ktoré časti incident reportu sú skutočné a ktoré sú vyvodené z výsledkov.

Referencie

(Pre úroveň poznania študijných cieľov)

Anderson, L. W. and Krathwohl, D. R. (eds) (2001) *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*, Allyn & Bacon

10. Príloha C – Pravidlá používané pre základnú učebnú osnovu ISTQB

Tu uvedené pravidlá boli použité pri vývoji a revidovaní týchto osnov. (ZNAČKA je zobrazovaná po každom pravidle ako skrátenejší názov pravidla.)

10.1.1 Všeobecné pravidlá

- ŠP1. Učebná osnova by mala byť zrozumiteľná a absorbovateľná ľuďmi so žiadnou až šesťmesačnou (alebo dlhšou) skúsenosťou v testovaní. (ŠESTMESAČNÝ)
- ŠP2. Učebná osnova by mala byť viac praktická ako teoretická. (PRAKTICKÝ)
- ŠP3. Učebná osnova by mala byť jasná a jednoznačná pre im určených čitateľov. (JASNÝ)
- ŠP4. Učebná osnova by mala byť zrozumiteľná ľuďom z rôznych krajín a ľahko preložiteľná do iných jazykov. (PRELOŽITEĽNÝ)
- ŠP5. Učebná osnova v origináli by mala používať americkú angličtinu. (AMERICKÁ ANGLIČTINA)

10.1.2 Aktuálny obsah

- ŠO1. Učebná osnova by mala zahŕňať súčasné koncepty testovania a mala by odrážať súčasné najlepšie praktiky v testovaní softvéru tam, kde je to všeobecne odsúhlasené. Osnova je predmetom revízie každých tri až päť rokov. (AKTUÁLNY)
- ŠO2. Učebná osnova by mala minimalizovať obsah záležitostí meniacich sa časom ako sú napríklad aktuálne trhové podmienky, aby mohla mať životnosť tri až päť rokov. (ŽIVOTNOSŤ)

10.1.3 Študijné ciele

- ŠC1. Študijné ciele by mali rozlišovať medzi bodmi na rozpoznanie/zapamätanie (úroveň znalostí Z1), bodmi, ktorým by kandidáti mali rozumieť koncepčne (Z2), bodmi, ktoré by kandidáti mali byť schopní používať v praxi (Z3) a bodmi, ktoré by mali byť kandidáti schopní použiť k analýze dokumentu, softvéru alebo situácie na projekte v danom kontexte (K4). (ZNALOSTNÁ ÚROVEŇ)
- ŠC2. Popis obsahu by mal byť konzistentný s cieľmi štúdia. (KONZISTENÝ SO ŠC)
- ŠC3. K objasneniu študijných cieľov by mali byť vydané spolu s učebnou osnovou vzorové skúšobné otázky pre každú hlavnú časť. (OTÁZKY K ŠC)

10.1.4 Celková štruktúra

- ŠŠ1. Štruktúra osnov by mala byť jasná a mala by dovoliť krížové odkazy na iné a tiež z iných častí, referenciu zo skúšobných otázok a z ďalších relevantných dokumentov. (KRÍŽOVÉ ODKAZY)
- ŠŠ2. Prekrývanie medzi časťami osnov by malo byť minimalizované. (PREKRYTIE)
- ŠŠ3. Všetky časti osnov by mali mať rovnakú štruktúru. (KONZISTENCIA ŠTRUKTÚRY)
- ŠŠ4. Osnovy by mali obsahovať verziu, dátum vydania a číslo strany na každej strane. (VERZIA)
- ŠŠ5. Osnovy by mali zahrňovať smernice pre množstvo času, ktoré je potrebné venovať každej časti (vyjadriť úmernú dôležitosť každej témy). (STRÁVENÝ ČAS)

Referencie

- ŠR1. Zdroje a referencie budú uvedené pre koncepty v osnovách aby tak pomohli poskytovateľom školení nájsť dostatok informácií o téme. (REFERENCIE)

ŠR2. V prípade, ak nie sú pripravené identifikované a jasné zdroje, malo by byť v osnovách uvedených viacej detailov. Napríklad definície sú v glosári, takže v osnovách sú uvedené len pojmy. (DETAILY BEZ REFERENCIÍ)

Zdroje informácií

Pojmy použité v osnovách sú definované vo výkladovom slovníku ISTQB používaných pri testovaní softvéru. Aktuálna verzia glosára je dostupná v ISTQB(anglický jazyk) alebo CaSTB (slovenský jazyk).

Zoznam odporúčanej literatúry o testovaní softvéru je taktiež vydaný paralelne s týmito osnovami. Hlavný zoznam literatúry je uvedený v časti Referencie.

11. Príloha D – Upozornenie poskytovateľom školení

Každému hlavnému nadpisu v osnovách je pridelený vyznačený čas v minútach. Účelom je odporučiť proporcionálne rozdelenie času pre každú časť v rámci akreditovaného školenia a zároveň určiť približné minimum času pre výuku daných častí.

Poskytovatelia školení môžu danej časti venovať viac času ako sa uvádza v osnovách a taktiež kandidáti môžu venovať viac času čítaniu a štúdiu. Študijný plán školenia nemusí presne kopírovať poradie uvedené v osnovách.

Osnovy obsahujú referencie na platné štandardy, ktoré musia byť použité pri príprave školiaceho materiálu. Každý použitý štandard musí byť vo verzii špecifikovanej v aktuálnej verzii týchto osnov. Ďalšie publikácie, šablóny alebo štandardy, na ktoré nie sú referencie v tejto učebnej osnove, môžu byť taktiež použité, avšak nebudú predmetom skúšania.

Všetky študijné ciele Z3 a Z4 vyžadujú, aby boli do študijných materiálov zahrnuté praktické cvičenia.

12. Príloha E – Poznámky k vydaným verziám

Vydanie z roku 2010

1. Zmeny študijných cieľov (ŠC) zahŕňajú niektoré upresnenia
 - a. Zmenené znenie pre nasledovné ŠC (obsah a stupeň ŠC zostávajú nezmenené): ŠC-1.2.2, ŠC-1.3.1, ŠC-1.4.1, ŠC-1.5.1, ŠC-2.1.1, ŠC-2.1.3, ŠC-2.4.2, ŠC-4.1.3, ŠC-4.2.1, ŠC-4.2.2, ŠC-4.3.1, ŠC-4.3.2, ŠC-4.3.3, ŠC-4.4.1, ŠC-4.4.2, ŠC-4.4.3, ŠC-4.6.1, ŠC-5.1.2, ŠC-5.2.2, ŠC-5.3.2, ŠC-5.3.3, ŠC-5.5.2, ŠC-5.6.1, ŠC-6.1.1, ŠC-6.2.2, ŠC-6.3.2.
 - b. ŠC-1.1.5 bol preformulovaný a povýšený na Z2, nakoľko môže byť očakávané porovnanie termínov súvisiacich s termínom defekt.
 - c. Bol pridaný ŠC-1.2.3 (Z2). Obsah bol zahrnutý už v učebnej osnove z roku 2007.
 - d. ŠC-3.1.3 (Z2) teraz kombinuje obsah ŠC-3.1.3 a ŠC-3.1.4.
 - e. ŠC-3.1.4 bol z učebnej osnovy 2010 odstránený, nakoľko je čiastočne duplicitný s ŠC-3.1.3.
 - f. ŠC-3.2.1 bol preformulovaný z dôvodu udržania konzistencie s obsahom učebnej osnovy 2010.
 - g. ŠC-3.3.2 bol upravený a jeho stupeň zmenený zo Z1 na Z2, z dôvodu udržania konzistencie so ŠC-3.1.2.
 - h. ŠC-4.4.4 bol upravený kvôli zrozumiteľnosti a bol zmenený zo Z3 na Z4. Dôvod: ŠC-4.4.4 už bol napísaný v štýle Z4.
 - i. ŠC-6.1.2 (Z1) bol vypustený z učebnej osnovy verzie 2010 a bol nahradený ŠC-6.1.3 (Z2). V učebnej osnove 2010 nie je ŠC-6.1.2.
2. Konzistentné používanie pre prístup k testovaniu, v súlade s definíciou vo výkladovom slovníku. Výraz stratégia testovania nebude vyžadovaná ako výraz na ktorý je potrebné si spomenúť.
3. Kapitola 1.4 teraz obsahuje koncept sledovateľnosti medzi základom testovania a testovacími prípadmi.
4. Kapitola 2.x teraz obsahuje ciele testovania a základ testovania.
5. Hlavným pojmom podľa výkladového slovníka je teraz retestovanie a nie konfirmačné testovanie.
6. Na viacerých miestach učebnej osnovy bol pridaný aspekt kvality dát a testovania: kvalita dát a riziko v kapitole 2.2, 5.5, 6.1.8.
7. Ako nova podkapitola bola pridaná kapitola 5.2.3 Vstupné kritériá. Dôvod: súlad s ohľadom na výstupné kritériá (-> vstupné kritériá pridané do ŠC-5.2.9).
8. Konzistentné používanie stratégie testovania a prístupu k testovaniu s ich definíciami vo výkladovom slovníku.
9. Kapitola 6.1 bola skrátená nakoľko popisy nástrojov boli pre 45 minútovú vyučovaciu hodinu veľmi obsiahle.
10. Bol vydaný standard IEEE Std 829:2008. Táto verzia učebnej osnovy ešte na toto nové vydanie neberie ohľad. Kapitola 5.2 sa odvoláva na dokument hlavný testovací plán. Obsah hlavného testovacieho plánu je založený na koncepte, že document "testovací plán" zahŕňa rôzne úrovne plánovania: testovacie plány pre jednotlivé testovacie úrovne môžu byť vytvorené, taktiež ako aj testovací plán na projektovej úrovni, ktorý zahŕňa viaceré testovacie úrovne. Posledne menovaný sa v učebnej osnove ako aj vo výkladovom slovníku ISTQB nazýva hlavný testovací plán.
11. Etický kódex bol presunutý z CTAL do CTFL.

Vydanie z roku 2011

Vykonané zmeny vydaním údržbovej verzie 2011

1. Všeobecne: termín Working Party bol nahradený termínom Working Group (pracovná skupina).
2. Pojem post-conditions (výstupné podmienky) bol vymenený za pojem postconditions z dôvodu konzistentnosti s výkladovým slovníkom ISTQB v2.1.
3. Prvý výskyt: ISTQB nahradené ISTQB@.
4. Úvod k učebnej osnove: Popis kognitívnych stupňov znalostí bol odstránený, nakoľko bol duplicitný s popisom v prílohe B.
5. Sekcia 1.6: Keďže nebolo zamýšľané definovať študijný cieľ pre "Etický kódex", bol odstránený kognitívny stupeň pre túto sekciu.
6. Sekcia 2.2.1, 2.2.2, 2.2.3 a 2.2.4, 3.2.3: Opravené problem s formátovaním v zoznamoch.
7. Sekcia 2.2.2: Slovo zlyhanie nebolo správne pre "... izolovať zlyhania špecifického komponentu ..." Preto bolo v tejto vete nahradené pojmom "defekt".
8. Sekcia 2.3: Bolo opravené formátovanie odrážkového zoznamu cieľov testovania, ktoré súvisia s termínmi testovania v sekcii Typy testov (Z2).
9. Sekcia 2.3.4: Aktualizovaný popis pojmu ladenie, tak aby bol v súlade s výkladovým slovníkom ISTQB v2.1.
10. V sekcii 2.4 bolo z časti "zahŕňa testovanie údržby aj rozsiahle regresné testovanie" odstránené slovo "rozsiahle", nakoľko "rozsiahle" je závislé na zmene (veľkosť, riziká, hodnota, atď.), tak ako je to popísané v ďalšej vete.
11. Sekcia 3.2: Za účelom objasnenia vety bolo odstánené slovo "zahrnutím".
12. Sekcia 3.2.1: Keďže aktivity formálnej revízie boli zle naformátované, revízny process dostal 12 hlavných aktivít miesto šiestich, ako bolo zamýšľané. Bolo to zmenené späť na šesť, čím sa dosiahlo to, že táto sekcia je v zhode s učebnou osnovou v2007 a učebnou osnovou pre pokročilý stupeň v2007.
13. Sekci 4: Slovo "vyvinutý" bolo nahradené slovom "definovaný", pretože testovacie prípady sa definujú a nie vyvíjajú.
14. Sekcia 4.2: Zmena textu za účelom objasnenia toho, ako môže byť testovanie čiernej a bielej skrinky použité spoločne s technikami založenými na skúsenostiach.
15. Sekcia 4.3.5 zmena textu "... medzi aktérmi, vrátane užívateľov a systému ..." na "... medzi aktérmi (užívateľmi alebo systémami), ... "
16. Sekcia 4.3.5 pojem alternatívna cesta nahradená pojmom alternatívny scenár.
17. Sekcia **Error! Reference source not found.**:2: Z dôvodu vyjasnenia pojmu testovanie vetvenia v texte sekcie **Error! Reference source not found.**, bola zmenená veta, ktorá objasňuje zameranie testovania vetvenia.
18. Sekcia 4.5, sekcia 5.2.6: Pojem "experienced-based" bol nahradený správnym termínom "experience-based" (založený na skúsenostiach).
19. Sekcia 6.1: Nadpis "6.1.1 Pochopenie významu a účelu podporných nástrojov pre testovanie (Z2)" bol nahradený nadpisom "6.1.1 Podporné nástroje pre testovanie (Z2)".
20. Sekcia 7 / Literatúra: Uvedená tretia edícia [Black,2001], nahradzujúca druhú edíciu.
21. Príloha D: Kapitoly vyžadujúce si cvičenia boli nahradené všeobecnou požiadavkou, že všetky študijné ciele úrovne Z3 a vyššie vyžadujú cvičenia. Toto je požiadavka špecifikovaná v Akreditačnom procese ISTQB (verzia 1.26).
22. Príloha E: Zmenené študijné ciele medzi verziami 2007 a 2010 už sú vymenované správne.

13. Index

akceptačné testovanie na mieste	24	konfirmačné testovanie ..	13, 15, 16, 19, 25, 26
akčné slová	63	kontrolné zoznamy	30, 31, 58
aktivity plánovania testovania	47	krabicový softvér	21
alfa testovanie	22, 24	kvalita 8, 10, 11, 12, 13, 18, 25, 34, 36, 45, 46, 48, 52, 54, 58	
analýza hraničných hodnôt	34, 38	ladenie	13, 22, 26, 57, 61
analýza testovania	15, 36, 46, 47	manažér testovania	8, 45, 46, 52
architektúra	15, 19, 21, 23, 25, 26	metrika	30, 31, 43
archivovanie/archivácia	16, 27	miera zlyhaní	48, 49
automatizácia	26	modelovací nástroj	59
beta tesovanie	22,24	modely softvérového vývoja	20
bezpečnosť	24, 25, 33, 45, 48, 57, 60	modely vývoja	19, 20
cieľ testovania	13, 20, 25, 41, 42, 46, 49	moderátor	30, 31
ciele pre testovanie	19	moderátor/líder inšpekcie	30
dátový tok	33	monitorovací nástroj	45, 57, 60
defekt 10, 11, 12, 13, 14, 16, 17, 19, 22, 23,25, 26, 28, 29, 30, 31, 32, 33, 35, 37, 38, 39, 41, 42, 43, 45, 47, 48, 49, 52, 53, 54, 57, 58, 59, 60, 69		“nahraj/reprodukuje” nástroje	59
dopadová analýza	19, 27, 36	nahratý skript	62
dynamické testovanie	13, 28, 29, 33, 58	následné kroky	30, 31
faktory úspechu	32	nástavec	22, 59, 60
formálna revízia	28, 30	nástroj na prípravu testovacích dát	55, 57
framework jednotkového testovania	22, 57, 59, 60	nástroj na riadenie testovania	57, 63
funkcionalita	22, 23, 25, 47, 52, 62, 64	nástroj na vykonanie testu 16, 36, 56, 57, 59, 60, 62	
funkcionálna špecifikácia	25	nástroj podporujúci iný nástroj	61
funkcionálna úloha	23	nástroj pre bezpečnosť	57, 60
funkcionálne požiadavky	22, 23	nástroj pre dynamickú analýzu	57, 60
funkcionálne testovanie	25	nástroj pre framework jednotkového testovania 57, 59, 60	
funkcionálny test	25	nástroj pre konfiguračný manažment	57, 58
hustota defektov	47, 49	nástroj pre ladenie	22, 57, 61
chyba	10, 11, 41, 59	nástroj pre meranie pokrytia	57, 60
implementácia testovania	15, 16, 36, 47	nástroj pre navrhovanie testu	57, 59
incident ..	15, 16, 18, 22, 44, 46, 54, 57, 58, 62	nástroj pre podporu testovania	56, 62
inkrementálny vývojový model	20	nástroj pre revidovanie	57, 58
inšpekcia	28, 30, 31, 32	nástroj pre riadenie incidentov	57, 58
integrácia/integračný 13, 20, 21, 22, 23, 26, 33, 38, 39, 40, 43, 46, 57, 60, 69		nástroj pre riadenie požiadaviek	57, 58
integračné testovanie 20, 21, 22, 23, 26, 33, 38, 43, 57, 60, 69		nástroj pre sledovanie defektov	57, 58
integračné testovanie komponentov 20, 22, 26, 57, 60		nástroj pre stres testovanie	57, 60
ISO 9126	11, 25, 27, 65	nástroj pre testovanie výkonu	57, 60, 63
kick-off	30	nástroj pre záťažové testovanie	57, 60
klasifikácia testovacích nástrojov	57	nástroj statickej analýzy 28, 33, 57, 58, 59, 60, 63	
klúčové osoby	12, 13, 16, 17, 23, 37, 43, 53	nástroje podporujúce špecifické oblasti aplikácií	60
kompilátor	33	nástroje pre podporu statického testovania .	58
komplexnosť	11, 33, 48, 58	nástroje pre podporu špecifikácie testov	59
konfiguračný manažment	43, 46, 51, 57	nástroje pre podporu vykonania testov a zaznamenávanie	59
		návrh testov 13, 15, 20, 34, 35, 36, 37, 41, 46, 48, 57, 59, 62	

neformálna revízia	28, 30, 31	revidovanie/revízia	13, 18, 28, 29, 30, 31, 32, 33, 45, 46, 52, 54, 57, 58, 67, 70
nefunkcionálne požiadavky	19, 22, 23	revidujúci	30, 31
nevýhody nezávislosti	45	revízia kolegami na rovnakej úrovni	30, 31
nezávislosť	17, 45, 46	riadenie incidentov	46, 54, 57
núdzová zmena	27	riadenie testovania	15, 43, 49, 50
očakávaný výsledok	16, 34, 36, 46, 63	riadenie testovania	43, 57, 58
odhadovanie omylov	17, 41, 48	riadenie verzií	51, 57
odhadovanie testovania	48	riadiace nástroje	46, 57, 58, 63
omyl	10, 11, 17, 41, 48	riadiace toky	25, 33, 34, 40
oprava	27	riziká	11, 22, 47, 52
organizácia testovania	45	riziká používania nástrojov	62
osobitné úvahy ku niektorým typom nástrojov	62	riziko	11, 12, 13, 14, 22, 23, 26, 27, 35, 36, 42, 43, 47, 48, 50, 52, 53, 58
ovládač	22, 59	roly	8, 28, 30, 31, 32, 45, 46, 47
pesticídny paradox	14	rozdelenie ekvivalencie	34, 38
plán vykonania testu	36	RUP (Rational Unified Process)	20
plánovanie testovania	15, 16, 43, 47, 51, 53, 73	sekvencie spracovania transakcií	23
podmienky testovania	13, 15, 25, 36, 37	simulátory	22
podnikové akceptačné testovanie	24	skriptovací jazyk	59, 62, 63
podpora nástrojov	22, 29, 40, 56, 62	sledovanie postupu testovania	49
podporné nástroje pre riadenie testovania a testov	57	sledovanie testu	46, 49, 50
podporné nástroje pre výkon a monitorovanie	60	sledovateľnosť	34, 36, 46, 51, 57, 58, 60
pokrytie kódu	25, 26, 34, 40, 57	softvér vyvinutý na zákazku	24
pokrytie príkazov	40	softvérový vývoj	8, 10, 11, 19, 20
pokrytie rozhodnutí	34, 40	spoľahlivosť	11, 13, 25, 47, 48, 52, 57
pokrytie testovania	15, 47, 48	statická analýza	29, 33
pokrytie	15, 22, 25, 26, 34, 36, 37, 38, 40, 47, 48, 49, 57, 58, 60, 62	statická technika	28, 29
použitelnosť	11, 24, 25, 43, 45, 52	statické testovanie	13, 29
požiadavka	13, 20, 22, 29, 31, 57, 58	stres testovanie	25, 57, 60
prevádzkové akceptačné testovanie	24	sumárna správa testovania	15, 16, 43, 46, 49
prevádzkové testovanie	13, 21, 27	systémové integračné testovanie	20, 22
prieskumné testovanie	41, 48, 59	systémové tesovanie	13, 20, 22, 23, 24, 47, 69
princípy testovania	10, 14	špecifikácia požiadaviek	23, 25
prípady použitia	20, 23, 25, 39	špecifikácia testovacej procedúry	34, 36
prístup k testovaniu	36, 46, 47, 48, 49	špecifikácia testovacieho prípadu	34, 36, 54
prístup RAD (Rapid application development)	20	špecifikácie návrhu testov	43
prístup riadenia dátami	63	štruktúrne testovanie	22, 25, 26, 40
prístup riadený kľúčovými slovami	63	študijné ciele	8, 9, 10, 19, 28, 34, 43, 56, 69, 70
prístup založený na rizikách	53	technická revízia	28, 30, 31, 32
proces vývoja testu	36, 73	technika čiernej skrinky	34, 37, 38
produktové riziko	17, 43, 52, 53	technika návrhou testov	34, 35, 36, 37
projektové riziko	12, 43, 52, 53	technika návrhu testovania založená na špecifikácii	37
protokol o teste	15, 16, 41, 59	technika návrhu testovania založená na štruktúre	37, 40
prototypovanie	20	technika návrhu testu bielej skrinky	37, 40
regresné testovanie	15, 16, 19, 25, 26, 27	technika návrhu testu čiernej skrinky	37
regulačné akceptačné testovanie	24	technika návrhu testu založená na skúsenostiach	37
report testovania	43, 49	Technika založená na skúsenostiach	35, 37, 41
reportovanie z testovania	43, 49	technika založená na špecifikácii	26, 37, 38
retestovanie	26, pozri konfirmačné testovanie		

techniky založené na štruktúre	37, 40
test prípadu použitia	34, 38
tester 10, 13, 17, 30, 35, 39, 41, 43, 45, 46, 51, 62, 67	
testovací plán 15, 16, 29, 43, 46, 47, 51, 52, 53, 73	
testovací postroj	16, 22, 51, 57, 59
testovací případ 13, 14, 15, 16, 22, 25, 29, 34, 35, 36, 37, 38, 39, 40, 43, 49, 54, 59, 69	
testovací skript	16, 29, 36
testovacia procedúra	15, 16, 34, 36, 43, 47
testovacia prognóza	59, 60
testovacia stratégia	15, 46, 47, 48
testovacie dáta 15, 16, 36, 38, 46, 57, 59, 62, 63	
testovacie prostredie 15, 16, 22, 23, 46, 49, 50, 59	
testovacie úsilie	48
testovanie "v teréne"	22, 24
testovanie a kvalita	11
testovanie bezpečnosti	25, 60
testovanie bielej skrinky	25, 40
testovanie čiernej skrinky	25
testovanie komponentov 20, 22, 24, 26, 34, 39, 40	
testovanie na prvom mieste	22
testovanie použiteľnosti	25, 43
testovanie prechodu stavov	34, 38, 39
testovanie prenositeľnosti	25
testovanie príkazov	34, 40
testovanie prípadov použitia	34, 38, 39
testovanie riadené dátami	62
testovanie riadené kľúčovými slovami	62
testovanie robustnosti	22
testovanie rozhodovacej tabuľky	34, 38, 39
testovanie rozhodovaní	34, 38
testovanie rozpoľahlivosti	25
testovanie spolupôsobenia	25
testovanie údržby	19, 27
testovanie udržovateľnosti	25
testovanie výkonu	25, 26, 57, 60, 63
testovanie založené na rizikách	48, 52, 53
testovanie založené na špecifikácii . 25, 34, 35	
testovanie založené na štruktúre	34, 40
testvér	15, 16, 46, 51, 58, 60
typy testov	19, 25, 27, 46
typy testovacích nástrojov	56, 57
úlohy testera	46
úlohy vedúceho testovania	46
upgrade	27
úroveň testovania 19, 20, 22, 25, 26, 27, 34, 38, 40, 42, 43, 46, 47	
útok na chyby	41
uzavretie testu	10, 15, 16
užívateľské akceptačné testovanie	22, 24
validácia	20
vedúci testovania	17, 43, 45, 46, 54
verifikácia	20
V-model	20
vnorené systémy	60
vplyv skúšania	57
vstupné kritéria	30
výber testovacích techník	42
vyčerpávajúce testovanie	14
výhody nezávislosti	45
výhody používania nástrojov	62
vykonanie testu 13, 15, 16, 29, 33, 36, 41, 43, 56, 57, 59	
vypelosť	16, 30, 36, 63
výstupné kritéria 13, 15, 16, 30, 31, 43, 46, 47, 49	
vývoj 8, 10, 11, 12, 13, 14, 17, 19, 20, 22, 23, 26, 29, 30, 33, 36, 42, 45, 47, 48, 51, 52, 54, 59, 60, 67	
vývoj riadený testovaním	22
walkthrough	28, 30, 31
základ testovania	15
základná príčina	10, 11
zapisovateľ	30, 31
záťažové testovanie	25, 57, 60
zavedenie nástroja v organizácii	56, 64
záznam o incidente	44, 54
zaznamenávanie incidentov	54
zaznamenávateľ	30
zdola-nahor	23
zhora-nadol	23
zlepšenie	24, 27
zlyhanie 10, 11, 13, 14, 17, 19, 22, 23, 29, 33, 41, 44, 48, 49, 52, 53, 58, 59, 69	
Zmluvné akceptačné testovanie	24
Zodpovednosti	22, 28, 30
zostava testovania	26