

# Testovanie biela a čierna skrinka

MARTIN NEPŠINSKÝ

*Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
martin.nepsinsky[zavináč]gmail[.]com*

**Abstrakt.** Táto esej sa zaoberá zabezpečením kvality v softvérovom projekte a to konkrétne jednou zo základných fáz životného cyklu softvérového projektu, testovaním. Poznáme niekoľko spôsobov delenia testovaní, ale v tejto práci sa venujem iba jednému z najznámejších a to deleniu na testovanie biela a čierna skrinka. Pri väčšine projektov je potrebné v rôznej miere vykonávať oba spomínané typy testovaní, ale manažéri musia dbať na ich rozdiely a prispôbiť tomu plánovanie a riadenie v softvérovom projekte. Táto práca preto stručne špecifikuje oba prístupy a snaží sa ich porovnať s pohľadom manažéra projektu a z pohľadu rôznych usporiadaní a zložení tímov.

## Úvod

V tomto dokumente som sa zameral najmä na manažment zabezpečenia kvality v softvérovom projekte. Kvalita môže byť definovaná ako stupeň splnenia požiadaviek a potrieb používateľa, ale väčšinou si pod týmto pojmom predstavíme správnosť, efektívnosť, spoľahlivosť a bezchybnosť. Spoločnosti sa snažia znížiť čas potrebný na vývoj softvéru, zvýšiť kvalitu a zefektívniť údržbu systémov. Na vývoj softvéru sa kladú neustále väčšie nároky nielen z hľadiska rýchlosti vývoja a nákladov, ale aj kvality. Softvérové systémy sú komplexnejšie, zložitejšie a môžu sa používať dlhší čas, čím rastie čas vynaložený na údržbu systému. Podľa [3] si zabezpečenie požadujúcej kvality softvérového produktu môže vyžadovať až 50% prostriedkov a v prípade ak je požadovaná vysoká miera bezpečnosti, stability a správnosti systému, ako napríklad v zdravotníctve alebo letectve, tak to môže byť aj viac ako 50% nákladov vynaložených na zabezpečenie kvality.

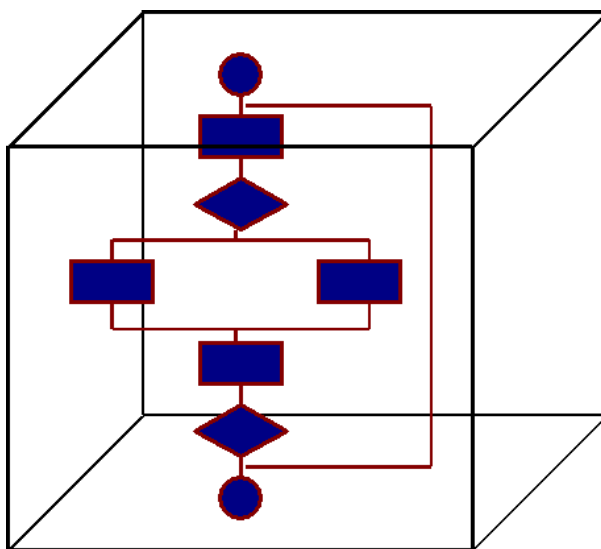
Jedným zo spôsobov ako zabezpečiť vyššiu kvalitu softvéru je zamerať sa na kvalitu procesu vývoja softvéru. Pretože v softvérovom inžinierstve nie je jednoduché merať kvalitu produktov, často sa berie ohľad práve na proces vývoja. Organizácie sa snažia získať certifikáty štandardov rôznych organizácií preukázaním, že ich proces vývoja vyhovuje daným štandardom.

Ďalší zo spôsobov ako dosiahnuť vyššiu kvalitu softvéru je dôkladné testovanie systému. Pri testovaní sa vývojári a testeria snažia identifikovať a odstrániť čo najviac odchýliek od očakávaného správania systému.

V mojej práci som sa venoval dvom z mnohých spôsobov delení testovaní a to testovaniu biela skrinka (white box) a čierna skrinka (black box) so zameraním na manažment, plánovanie, riadenie a procesy prebiehajúce v organizácií.

## Biela skrinka

Testovanie biela skrinka sa zameriava na vnútornú logiku a štruktúru systému, preto sa niekedy označuje aj ako štrukturálne. Testovaný systém si môžeme predstaviť ako priehľadnú skrinku, do ktorej vidíme a poznáme všetky jej súčasti. Testovanie biela skrinka sa zakladá na dôkladnej znalosti implementácie systému. Pretože vyžaduje znalosť kódu aplikácie, môže prebiehať okamžite ako je vytvorená testovaná časť kódu. Pri tomto testovaní sa môže vykonávať analýza toku dát, toku riadenia, štruktúry kódu a spracovanie výnimiek a chybových stavov na otestovanie požadovaného správania systému.

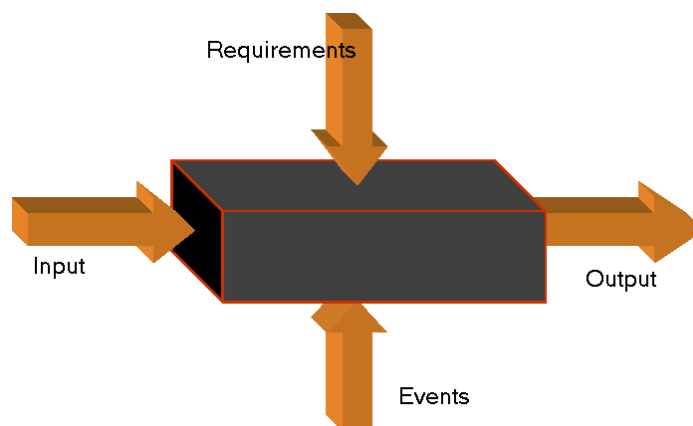


Obr. 1. Testovanie biela skrinka [2]

## Čierna skrinka

Pri testovaní čierna skrinka tester vychádza zo špecifikácie alebo požiadaviek systému, bez vedomosti o vnútornej štruktúre. Testovaný systém predstavuje čiernu skrinku,

ktorá je určitým spôsobom nakonfigurovaná a prijíma vstupy, väčšinou cez používateľské rozhranie alebo webový prehliadač. Tester skúša zadávať rôzne vstupy a sleduje výstupy, ktoré porovnáva z očakávanými výsledkami.



**Obr. 2.** Testovanie čierna skrinka [2]

Existuje aj testovanie šedá skrinka (grey box), ale pretože ide o určitú kombináciu techník testovaní biela a čierna skrinka, tak sa mu nebudem ďalej venovať.

## Porovnanie testovaní

Oba spôsoby testovania sa väčšinou skladajú z rovnakých fáz, ale v rámci každej fázy sa podstatne odlišujú. Pri efektívnom manažovaní projektu sa musí klásť dôraz na tieto rozdiely a prispôbiť im plánovanie a riadenie testovania. Pri každom type testovania sú navyše potrebné aj odlišné zdroje a je rôzna viditeľnosť výsledkov, čo treba zohľadniť pri hodnotení a dokumentovaní projektu.

Aby mohlo testovanie úspešne začať je potrebné zabezpečiť rôzne zdroje pre projekt. Pre rôzne typy testovaní je potrebné zabezpečiť rôznych ľudí, na čo sa využíva manažment ľudských zdrojov. Pri testovaní biela skrinka sa vyžadujú od testerov oveľa vyššie znalosti ako pri testovaní čierna skrinka. Testerí sú väčšinou samotní vývojári, ktorí musia rozumieť kódu, architektúre systému, štruktúre dát alebo bezpečnostným opatreniam systému, poprípade iba časti, ktorú testujú. Na testovanie čierna skrinka sú potrební ľudia, ktorý sú schopný naučiť sa pracovať so systémom a je vhodné, aby časť z nich boli obyčajní používatelia. Existujú aj nástroje, ktoré dokážu automaticky vytvárať testy a testovať systém, čo ešte viac zníži potrebné zdroje. Niektoré takéto nástroje sú predstavené v práci [5]. Náklady na testovanie biela skrinka preto budú podľa mňa vyššie.

Pre testerov pomocou metódy biela skrinka musí manažér zabezpečiť, základnú požiadavku a to prístup k zdrojovému kódu. Pri väčších projektoch by mala byť zabezpečená aj analýza rizík.

Pre testerov pomocou metódy čierna skrinka stačí zabezpečiť prístup k systému, takí akí majú obyčajný používatelia. Z toho vyplývajú rôzne výhody tohto spôsobu testovania. Spoločnosť môže jednoduchšie využiť napríklad prácu z domu alebo využiť služby špeciálnej testerskej spoločnosti bez nutnosti poskytnúť im internú dokumentáciu k projektu. Klesajú aj náklady na softvérové vybavenie, pretože napríklad nie sú potrebné programy na prácu so zdrojovým kódom alebo dokumentáciou.

### Plánovanie

Plánovanie testovania je dôležitá súčasť softvérového projektu a úsilie naň vynaložené sa podľa mňa niekoľko násobne vráti, vďaka efektívnejšiemu priebehu projektu. Podľa [1] plánovanie redukuje neurčitost', zlepšuje výkonnosť, zabezpečuje lepšie porozumenie cieľom projektu a vytvára základňu pre sledovanie a riadenie práce.

Pri plánovaní testovania čierna skrinka by sa podľa mňa malo vychádzať z podrobných prípadov použitia systému. Každému prípadu použitia by sa mala priradiť priorita podľa dôležitosti alebo miery rizika a podľa toho je možné naplánovať čas, ktorý sa strávi testovaním jednotlivých prípadov použitia. Dôležitá je tak isto aj analýza rizík, ktorá pomáha identifikovať možné ohrozenia pre konkrétne prípady použitia, možné dopady na systém pri zneužití potencionálnych chýb v systéme a pravdepodobnosť, s ktorou dôjde k riziku. Ďalším dôvodom pre plánovanie je, že pri väčšine systémov by bo veľmi náročné otestovať všetky vstupy pre všetky prípady použitia a niekedy aj nemožné v stanovenom čase na projekt. Pri testovaní čierna skrinka sa bude postupovať podľa plánu, v ktorom bude určené koľko času je potrebné venovať jednotlivým prípadom použitia. V podrobnejšom pláne môžu byť vstupy rozdelené aj do určitých tried a presnejšie špecifikované koľko času sa bude testovať ktorá trieda vstupov.

Viditeľnosť je dôležitá pri riadení a plánovaní testovania, ale nie vždy je jednoduché zistiť koľko môže daná činnosť trvať a koľko času chýba do ukončenia. Pri testovaní biela skrinka sa značná časť zdrojov musí venovať štúdiu kódu, architektúry a iných potrebných zdrojov. Odhadnúť, koľko bude trvať štúdium je podľa mňa dosť náročné, pretože to výrazne závisí od schopnosti jednotlivca učiť sa. Podobne je to aj s viditeľnosťou, pretože je komplikované odmerať koľko sa už tester naučil a ako danej problematike rozumie. Pri testovaní metódou biela skrinka sa môžu vytvárať špeciálne testovacie triedy, ktoré automaticky otestujú požadovanú časť kódu. V pláne môžu byť špecifikované prípady použitia, pre ktoré sa vytvoria testovacie triedy a táto činnosť má aj celkom dobrú viditeľnosť, pretože je zrejme koľko testov už bolo vytvorených.

Podľa dokumentu [4] by stratégia testovania mala byť založená na analýze rizík. Cieľom analýzy rizík je určiť možné zraniteľnosti a ohrozenia systému a dopad, ktorý môžu spôsobiť. Podľa miery rizika sa priradia zdroje na otestovanie rizikových častí. Čím je väčšie riziko, tým sa viac zdrojov využije na jeho odstránenie. Podľa mňa by stratégia testovania mala vychádzať aj z podrobného diagramu prípadov použitia. Myslím si, že pri veľkých systémoch je komplikované analyzovať riziká pre každú komponentu. Týmto by mohlo dôjsť k podceneniu nejakého rizika.

### Vplyv topológie organizácie na testovanie

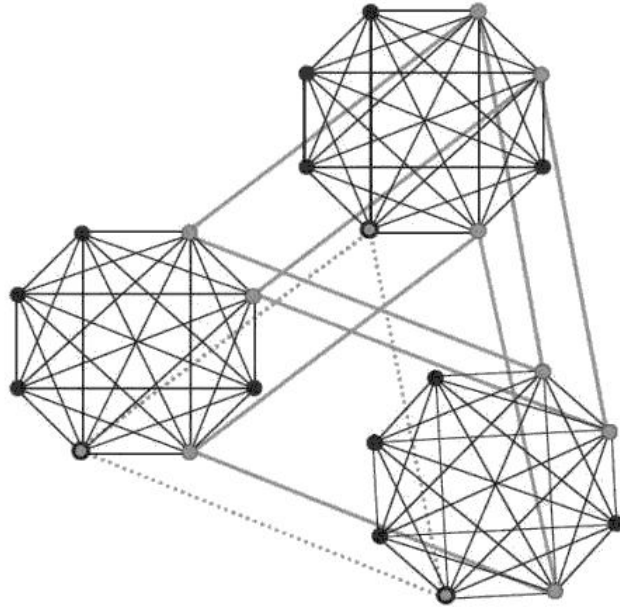
Pri manažovaní oboch typov testovaní, musia manažéri zohľadniť aj vplyv topológie organizácie na testovanie. Topológia organizácie môže podstatnou mierou ovplyvniť efektívnosť testovania. Pre rôzne typy testovaní je podľa mňa výhodnejšia iná topológia. V dokumente [1] boli identifikované štyri rôzne topológie, z ktorých každá má svoje výhody aj nevýhody pri použití testovania biela a čierna skrinka.

Pri funkcionálnej topológii sa miesta združujú na základe funkcionálnej špecializácie. To znamená, že všetci testerí aj vývojári budú mať svojho vedúceho skupiny a na projekte budú pracovať ľudia z rôznych skupín. Táto topológia je podľa mňa výhodnejšia pre testovanie čierna skrinka ako pre testovanie biela skrinka. Pri testovaní čierna skrinka sa testerí môžu zapojiť až vo fáze testovania systému a zatiaľ môžu pracovať na iných projektoch. Navyše môžu testovať metódou čierna skrinka aj viac systémov naraz, pretože táto metóda nevyžaduje žiadnu znalosť vnútornej štruktúry systému. Pri testovaní biela skrinka môže byť problém pri zdieľaní zdrojov, pretože tester sa musí dôkladne zoznámiť s požadovanými súčasťami a dokumentáciou systému.

Pri projektovej topológii je podľa mňa naopak efektívnejšie testovanie biela ako čierna skrinka. Pri tejto topológii sa združujú ľudia podľa príslušnosti k projektu. Nevýhodou testovania čierna skrinka je, že testerí ho môžu začať vykonávať až keď sa projekt dostane do fázy testovania, takže podstatnú časť životného cyklu nebudú efektívne využití. Výhodou pri testovaní biela skrinka je, že testerí sa môžu od začiatku zoznamovať so špecifikáciou a architektúrou systému. Pri tomto testovaní sa časť testov môže vytvoriť pred začiatkom fázy implementácie. Tento prístup je zreteľný napríklad pri metóde TDD (Test Driven Development) [7].

Na dôkladné otestovanie systému a zabezpečenie požadovanej kvality je väčšinou potrebné vykonať aj testovanie biela aj čierna skrinka. Z toho dôvodu by som pre testovanie nezvolil ani funkcionálnu ani projektovú topológiu, pretože každá je výhodnejšia len pre jednu metódu. V dokumente [1] boli ale predstavené aj iné topológie, ktoré sa zdajú byť vhodnejšie pre väčšie systémy.

Minimalizovať slabé a maximalizovať silné stránky predchádzajúcich topológií sa snaží maticová topológia, ale najvýhodnejšia sa mi zdá byť sieťová topológia (pozri Obr. 3), pri ktorej vznikajú projektové tímy ako zhluky alebo obežné dráhy, ktoré sa počas života projektu vyvíjajú. Tento prístup vyhovuje testovaniu čierna skrinka, pretože testerí sa môžu do projektu pripojiť až vo fáze testovania a navyše projektové tímy sa môžu prekrývať a zdieľať medzi sebou napríklad týchto testerov. Sieťová topológia vyhovuje aj testovaniu biela skrinka, pretože testerí sa môžu zaradiť do tímu už ku koncu fázy návrhu a môžu sa dôkladne venovať iba svojmu projektu.



Obr. 3. Sieťová topológia [6]

Pri menších tímoch, ako je napríklad náš tím na predmete Tvorba softvérového systému v tíme sa väčšinou využíva testovania biela popripade šedá skrinka. Je to z toho dôvodu, že všetci členovia tímu poznáme viac menej zdrojový kód, architektúru systému a štruktúru dát. Avšak niektoré tímy si mohli vyčleniť aj testera spôsobom čierna skrinka, ktorí nemal žiadne informácia o vnútornej štruktúre, čo je ale podľa mňa dosť nepravdepodobné. Pravdepodobnejšie je, že niektorí členovia tímu nepoznajú zdrojový kód, ale z dokumentácia poznajú napríklad architektúru systému, použité technológie alebo štruktúru a tok dát a v tom prípade ide o testovanie spôsobom šedá skrinka, čo je kombinácia dvoch porovnávaných spôsobov.

## Záver

Dôkladné testovania sa vráti počas údržby, dokonca podľa [3] môže stať opravenie chyby v nasadenom a spustenom systéme až dvadsať krát viac ako opravenie chyby počas testovania alebo implementácie.

Podľa mňa sa oba prístupy navzájom dopĺňajú, avšak pri plánovaní a riadení musia manažéri zohľadniť ich odlišnosť. V typickom prípade testovanie biela skrinka odhalí potencionálne zraniteľné časti systému a následne sa použije testovanie čierna skrinka na zistenie vstupov, ktoré spôsobujú neočakávané reakcie systému. Pri systémoch, u ktorých sa predpokladá určitá úroveň kvality by mali byť použité obe testovania na odstránenie maximálneho počtu odchýliek od očakávaného správania.

## Použitá literatúra

1. Bieliková, M.: Manažment v softvérovom inžinierstve, 1999.
2. Damian, D.: Practical Software Engineering: Software Testing, [cit. 2007-10-20], URL: <<http://sern.ucalgary.ca/courses/CPSC/451/W00/Testing.html>>
3. Gonzales, G., Parveen, T., Tilley, S.: A case study in test management, ACM Press, 2007.
4. Janardhanudu, G.: White Box Testing, Cigital Inc., 2005.
5. Michael, C.C., Radosevich, W.: BlackBox Security Testing Tools, Cigital Inc., 2005.
6. Nichols, W.R.: Building Successful Software Development Teams Using TSP and Effective Communication Networks, Bechtel Bettis Inc., 2006, [cit. 2007-10-20], URL: <<http://www.stsc.hill.af.mil/Crosstalk/2006/01/0601Nicols.html>>
7. Test-driven development, Wikipedia, the free Encyclopedia, [cit. 2007-10-20], URL: <[http://en.wikipedia.org/wiki/Test-driven\\_development](http://en.wikipedia.org/wiki/Test-driven_development)>.

## Annotation

### *Testing white and black box*

This essay focuses on ensuring the quality in software project and mainly on one of the phases of project's life cycle, testing. There are many types of testing division, but in this document we are focusing only at one of best known of them, division to white box testing and black box testing. In software project, in most cases it is required to perform both types of testing, but managers must pay attention to their differences and adjust planning and coaching in software project. This document shortly specifies both types of testing and tries to compare them from the view of managers and different topologies and composition of teams.