

QUALE

Quale 1/2014

Dane testowe - własny pomysł czy narzędzie?

Natalia Ziajka

Testować, czy nie testować?

Bartłomiej Prędki
Krzysztof Chytła

Jakość oprogramowania czy testowanie oprogramowania?

Wiktor Żołnowski

2014 z testowaniem

TestWell, BTD, TestingCup, TestWarez

Spis treści

INNY PUNKT WIDZENIA NA JAKOŚĆ

5. Jakość oprogramowania czy testowanie oprogramowania?
Wiktor Żołnowski

KRZYWYM OKIEM

9. Testować, czy nie testować?
Bartłomiej Prędki

11. Testować, czy nie testować?

Krzysztof Chytla

ZARZĄDZANIE

13. Współpraca z informatykami z Indii
Łukasz Smolarski

REDAKCJA

Redaktor naczelny
Karolina Zmitrowicz
karolina.zmitrowicz@quale.pl

Zastępca redaktora naczelnego:
Krzysztof Chytła
krzysztof.chytla@quale.pl

Redaktorzy:
Bartłomiej Prędki
bartlomiej.predki@quale.pl
Michał Figarski
michal.figarski@quale.pl
Piotr Poznański
piotr.poznanski@quale.pl

Website:
www.quale.pl/en (ENG)
www.quale.pl (PL)

Facebook:
http://www.facebook.com/qualemagazine

Spis treści

TESTOWANIE

21. Dane testowe - własny pomysł czy narzędzie?
Natalia Ziajka

35. Testowanie eksploracyjne
Tomasz Olszewski

39. W poszukiwaniu źródła innowacyjnych testerów
Maciej Zaborowski

WYDARZENIE

18. TestingCup 2014

20. Belgium Testing Days

32. Excited by Quality?

42. TestWarez2014

EDU - Edukacja IT dopasowana do czasów, w jakich żyjemy

Szkolenia informatyczne w Polsce nie nadążają za zmieniającą się kulturą i technologią. Niby mówią o rzeczach przełomowych, najnowszych osiągnięciach technologicznych, ale same tkwią w głębokiej przeszłości. Z trenerem mówiącym do statycznej tablicy, z uczestnikiem z papierowymi zeszytami. Ta prosta myśl doprowadziła do pomysłu przeniesienia szkoleń IT w XXI wiek.



EDU to zarówno metoda, jak i wirtualne miejsce stworzone po to, aby wspierać proces szkolenia się i przygotowania do szkoleń oraz egzaminów IT. Klasyczne szkolenia prowadzone przez trenera zostają rozszerzone o dodatkowy dostęp do materiałów elektronicznych, systemu powiadomień przed i po szkoleniu oraz systemu zarządzania wiedzą.

Funkcjonalność platformy jest rozwinięciem metody blended learning (B-learning), czyli metody kształcenia, która łączy tradycyjne metody uczenia się (bezpośredni kontakt z trenerem) z dodatkowymi ćwiczeniami i materiałami dostępnymi na komputerze (M-learning). Zaletą jest tu możliwość stosowania zarówno zdalnych, jak i bezpośrednich form komunikacji szkolących. Trener oraz uczestnicy mogą współpracować online zarówno przed, jak i na długo po szkoleniu.

EDU z jednej strony jest prostą platformą, gdzie można znaleźć materiały szkoleniowe zawsze w najbardziej aktualnej wersji. Z drugiej strony to złożony system komunikacji trener-uczestnik, pozwalający nie tylko pozyskać wiedzę, ale również nią zarządzać i utrzymać nawet na wiele lat po szkoleniu. System wewnętrznych powiadomień poinformuje uczestników o najnowszych wersjach. Dzięki niemu dowiedzą się również o zmianach i aktualizacjach w szkoleniach i egzaminach. Nie muszą więc pilnować zmian. EDU ich o tym poinformuje.

Co ważne, EDU dostarcza wszystkich informacji bez względu na to, gdzie jesteście. Nieważne czy logujecie się na komputerze, telefonie czy tablecie, zawsze sprawdzicie aktualne informacje o szkoleniu oraz skorzystacie z materiałów szkoleniowych.

Zapraszamy do zapoznania się z tym nowym na polskim rynku edukacyjnym rozwiązaniem: http://edu.ittraining.pl/o_edu

Wiktor Żołnowski

Jakość oprogramowania czy testowanie oprogramowania?



Ruszyliśmy! Początki są najważniejsze, więc w pierwszej odsłonie magazynu QUALE proponuję głębsze omówienie tematu zapewniania jakości oprogramowania (Quality Assurance), które wbrew temu jak wiele osób błędnie uważa nie jest tym samym, co testowanie oprogramowania.

Z czego wynika jakość oprogramowania? Czym charakteryzuje się oprogramowanie wysokiej jakości?

Według Wikipedii jakość oprogramowania to: „Ogół cech produktu programowego, które wpływają na zdolności spełniania przez niego określonych wymagań, m.in. elastyczność, funkcjonalność, integralność, niezawodność, efektywność, użyteczność, wydajność".

Według „Słownika wyrażeń związanych z testowaniem Wersja 2.0"[1] definicja jakości oprogramowania wygląda podobnie: „Ogół funkcjonalności i cech oprogramowania, które charakteryzują się zdolnością zaspokajania stwierdzonych lub przewidywanych potrzeb".

James Bach definiuje jakość oprogramowania w taki sposób: „Jakość jest wartością dla kogoś ważnego" [2] (Podobnego sformułowania użył wcześniej Jerry Weinberg - James dodał do niego stwierdzenie, że chodzi o kogoś ważnego - kogoś, kto się liczy, a nie zwykłą osobę).

Jakość niejedno ma imię. Kilka lat temu zadałem sobie pytanie, czy jakość oprogramowania można zapewnić jedynie poprzez jego testowanie? Lub raczej - czy testowanie jest wystarczające do zapewnienia jakości? Czy testowanie jest w ogóle konieczne, by zapewnić jakość oprogramowania?

Bez względu na definicję, którą sobie wybierzemy, testowanie może tylko w pewnym stopniu zweryfikować jakość wytwarzanego oprogramowania a niekoniecznie ją zapewnić. W tradycyjnym podejściu do wytwarzania oprogramowania, testowanie odbywa się przeważnie na końcu cyklu życia produktu. Współczesne metodyki (Agile/Lean) kładą nacisk nie tyle na zrównoleglenie procesu testowania z procesem kodowania, co połączenie tych procesów w jedno.

„Jakość jest wartością dla kogoś ważnego" - czy tester jest kimś ważnym? (Oczywiście, że tak, ale niekoniecznie w interesującym nas kontekście). Kto zatem jest wystarczająco ważny, by decydować o jakości wytwarzania oprogramowania? Klient? Osoba, która płaci za dostarczony software? Może tak… Ale jak wszyscy dobrze wiemy klienci przeważnie nie wiedzą, czego chcą, a jak już wiedzą - to jest to coś innego, niż chce rynek. Tak, właśnie „rynek" - to miejsce, gdzie ostatecznie trafia nasz produkt i gdzie jest używany (bądź nie) przez użytkowników.

To użytkownicy z pewnością są ważni na tyle, by stwierdzić, czy nasze oprogramowanie jest wysokiej jakości, czy nie. I raczej nie chodzi o pojedynczych użytkowników, ale o ogół osób, które korzystają z naszych produktów.

Jednym ze sposobów zapewniania jakości jest odpytywanie użytkowników. Nie jesteśmy oczywiście w stanie ankietować każdego z naszych użytkowników z osobna, pytając o to, czy dana funkcjonalność jest wysokiej jakości, lub czy produkt jest wysokiej jakości. Możemy jednak badać jakość na podstawie danych statystycznych. Możemy postawić hipotezę, że ludzie chętniej sięgają po produkty wysokiej jakości. Możemy założyć, że częściej używają produktów wysokiej jakości. W zasadzie w jakość produktu wpisuje się przecież jego użyteczność dla użytkowników.

Czy zatem wyrażanie jakości oprogramowania poprzez metryki pokazujące ilość zgłaszanych przez testerów błędów jest właściwą miarą dla realnej jakości? Czy testowanie wnosi cokolwiek do tematu jakości oprogramowania? Oczywiście, że tak! Gdyby nasz produkt był pełen bugów, to z pewnością użytkownicy prędzej czy później by z niego zrezygnowali, a straty wizerunkowe naszej marki bardzo szybko doprowadziłyby do pojawiania się nowych, skuteczniejszych konkurentów.

Spójrzmy jednak na ten problem z trochę innej perspektywy. Co jest nam potrzebne, by sprawnie i systematycznie weryfikować jakość naszego produktu? Potrzebujemy testować nasz produkt na realnych użytkownikach, śledzić i badać ich zachowanie oraz wyciągać odpowiednie wnioski. To jest właśnie podstawą wszelkich współczesnych, zwinnych metod wytwarzania oprogramowania spod znaku Agile i Lean Software Development. Wytwarzamy nasz produkt w sposób przyrostowy (dostarczając kolejne jego wersje funkcjonalność po funkcjonalności na produkcję) i badamy reakcję użytkowników na kierunek, w którym zrobiliśmy właśnie kolejny krok na drodze do stworzenia utopijnego „produktu idealnego". Każdy kolejny inkrement produktu jest eksperymentem - stawiamy sobie hipotezę, że funkcjonalność, którą zamierzamy w danej iteracji dostarczyć, będzie spełniała oczekiwania naszych użytkowników i podniesie szeroko rozumianą jakość produktu. Hipoteza ta jest następnie szybko weryfikowana przy użyciu metod statystycznych. W niektórych sytuacjach możemy potrzebować testować więcej, niż jedną hipotezę (lub jednoznacznie obalić naszą hipotezę przy użyciu testów negatywnych) - w takich sytuacjach powinniśmy skorzystać z testów A/B.

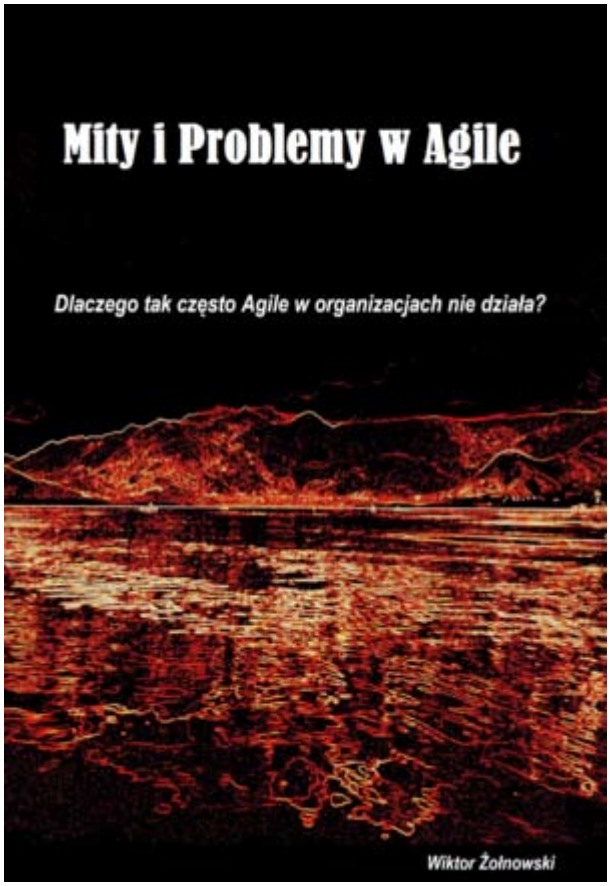
No dobrze, ale gdzie tutaj miejsce na zapewnianie jakości oprogramowania? Co gdybym powiedział, że rolą współczesnego specjalisty od Quality Assurance jest szukanie sposobów na to, by móc jak najszybciej weryfikować powyższe hipotezy? Nie tylko poprzez szukanie narzędzi do testów A/B, czy prowadzenie analiz

zachowań użytkowników. Można to robić także poprzez dobranie odpowiednich sposobów na to, aby oprogramowanie mogło być wytwarzane na tyle szybko, by łatwo było testować stawiane hipotezy. Najważniejsze jest zapewnienie elastyczności naszego oprogramowania, tak by mogło być ono zmieniane bez ryzyka wprowadzenia dużej ilości błędów, które nie zostaną wykryte. Czas wykrywania błędów oczywiście też ma znaczenie - kilkudniowe czy nawet kilkutygodniowe oczekiwanie na wyniki testów regresyjnych z pewnością nie sprzyja łatwemu i szybkiemu weryfikowaniu hipotez.

Automatyzacja testów jest dobrym przykładem na to, co możemy zrobić, by zapewnić jakość oprogramowania - nie dlatego, że testy automatycznie testują dokładniej, niż tester (co oczywiście nie jest nawet bliskie prawdy), ani nie dlatego, że takie testy są w stanie zastąpić kosztownych testerów. Testy automatyczne przyspieszają proces dostarczania informacji zwrotnej na temat błędów popełnianych podczas programowania, dzięki temu możemy programować szybciej i dostarczać do testów manualnych funkcjonalności już wstępnie zweryfikowane. Warto wspomnieć również o tym, że automatyzacja testów tylko na poziomie testów akceptacyjnych (pomimo, iż przyspiesza samo testowanie), nie zawsze sprawia, że jest ono tańsze. Ale to temat na kolejny artykuł.

Podsumowując proces zapewniania jakości ma wiele wspólnego z samym testowaniem oprogramowania, niemniej jednak testowanie oprogramowania samo w sobie jest tylko jednym z wielu elementów nad którymi powinniśmy pracować w celu zapewniania i podnoszenia jakości produktów.





Czy tester może być osobą odpowiedzialną za zapewnianie jakości? Tak, jeśli ma odpowiednią moc sprawczą w organizacji by (współ)decydować o stosowanych metodach wytwarzania oprogramowania.

Na zakończenie pozostawię Was jeszcze z otwartym pytaniem: Czy z poprawiania błędów w oprogramowaniu zawsze wynika poprawa jakości tegoż oprogramowania?
Jeśli będziecie mieli jakieś przemyślenia na ten temat, to zapraszam do dyskusji na facebooku i twitterze.

AUTOR

Wiktor Żołnowski



Wiktor Żołnowski - trener, konsultant, doradca, coach - głównie w obszarach testowania, automatyzacji testów, Agile, Scrum oraz Lean Startup. Autor książek o Agile i zmianie organizacji. W wolnych chwilach bloger piszący o jakości oprogramowania z trochę innego punktu widzenia.

Po pracy (którą traktuje raczej jak pasję niż regularną pracę) czasem udaje mu się wypłynąć w rejs po oceanie lub pograć na gitarze w jednym z krakowskich garażów.

Możecie go spotkać na licznych konferencjach i wydarzeniach w kraju i za granicą gdzie z chęcią dzieli się swoją wiedzą. Wiktor jest też twórcą i animatorem lokalnej społeczności Bring Your Own Problem oraz udziela się przy organizacji międzynarodowej konferencji Agile Lean Europe.

Książki: "Mity i Problemy w Agile" oraz "Agile Transformacje".

Blog: blog.testowka.pl

Szkolenia: agileszkolenia.pl

Bartłomiej Prędki

Testować, czy nie testować?

Oto jest pytanie

Testować, czy nie testować, oto jest pytanie. Witaj Szanowny Kolego. Jako, że jestem delikatnie bardziej posunięty w latach od Ciebie, naszło mnie ostatnio na egzystencjalne rozważania dotyczące naszej profesji. Wprawdzie może to wyglądać na podcinanie gałęzi, na której siedzimy, niemniej jednak zacząłem się zastanawiać, czy aby na pewno powinniśmy robić to, co robimy…

Jak zapewne wiesz, niemal na każdym szkoleniu z testowania, tłumaczy się uczestnikom, dlaczegóż to testowanie jest niezbędne, ważne i wartościowe, podając przy tym przykłady awarii oprogramowania, które okazały się opłakane w skutkach. Problem w tym, że w sporej części tych przykładów brak testów nie był najważniejszym czynnikiem prowadzącym do awarii. Wielu tych awarii udałoby się uniknąć, gdyby zarządzający projektami rzetelnie wykonali swoją robotę. Poniżej kilka przykładów pokazujących, że jednak lepiej, nie testować.

Powód pierwszy. Dlaczego na szkoleniach nie przywołuje się przykładów projektów, które odniosły spektakularny sukces i kluczowej roli testerów w osiągnięciu tego sukcesu? Bo to się słabo sprzedaje, poza tym testerzy psują soft – jak więc mogą być współtwórcami sukcesu? Wiadomo, że jeśli coś się schrzaniło, to lepiej jest powiedzieć „brak odpowiednich testów", niż „zarządzanie było słabe". Jeśli natomiast projekt kończy się wielkim sukcesem, to kto zbiera pochwały? Menedżerowie (bo świetnie zarządzali), programiści (bo świetnie napisali). Nikt wtedy nie pamięta o testerach – śmietankę spijają inni. I w ten oto sposób testowanie przedstawia się w negatywnych świetle – testerzy są po prostu potrzebni, żeby było na kogo zwalić winę za marne wyniki.

Powód drugi. Przyjrzyjmy się zarobkom programistów i testerów i pokażcie mi organizację, w której testerzy są wynagradzani lepiej, niż programiści. Nie widzę lasu rąk. Dlaczego? Bo takich organizacji NIE MA. Cała ta gadka o tym, jakie to testowanie jest „ważne", jest jedną wielką hipokryzją. Skoro więc testerzy mają pracować za garść orzeszków, to lepiej tę kasę przeznaczyć na porządne przeszkolenie programistów – nauczą się porządnie kodować i nie będą musieli znosić marudzenia tych, co nawet średniej krajowej nie potrafią zarobić... Menedżerowie też

będą zadowoleni – budżet na testy (o który nomen omen zawsze trzeba strasznie walczyć i tłumaczyć, „po co to komu") można też przeznaczyć na wydajniejsze komputery – szybciej się skompiluje, szybciej będzie można wypuścić release'a. Mamy więc lepszych programistów i wydajniejszy sprzęt – to po co nam wtedy testy?
Powód trzeci. Fakt: przetestowanie wszystkiego jest niemożliwe. Jeśli więc nie przetestujemy wszystkiego, to po co w ogóle się za to zabierać? Coś jakbym wsiadał do samochodu i od razu wiedział, że nie dotrę do wyznaczonego sobie celu. Albo włączając film, już na początku wiedziałbym, że go nie obejrzę do końca. Tak więc - robimy coś dobrze i do końca, albo dajemy sobie (i programistom) spokój. Żaden szanujący się Project Manager nie przyzna się klientowi, że system nie został przetestowany w 100%. Co więc robi? Prawdę tę ukrywa za kurtyną kolorowych wykresów, pokazując oczywiście tylko te, które wywołają na twarzy klienta błogi uśmiech zadowolenia. Coś mi tu tylko umknęło… Aha, że co – że to nieetyczne? Etyka? W BIZNESIE?

Powód czwarty. Ludzie lubią niską jakość. Wiele osób czerpie dziką satysfakcję z tego, że mają coś, co się psuje – bo sobie wtedy naprawią i pokażą, jacy to oni są zdolni. Inni znowu po prostu potrzebują od świata odrobiny współczucia z tego powodu, iż los pokarał ich takim badziewiem.

Powód piąty – i ostatni (więcej nie będzie, bo nie wiem, czy ktoś w tej sekcji magazynu przetestował poprawność wyświetlania się zbyt długim tekstów). Jak powiedział niejaki Edward Murphy: „Jeśli coś może się zepsuć, to zepsuje się na pewno".

I żadne testowanie tu nie pomoże.

AUTOR

Bartłomiej Prędki



Swoją karierę rozpocząłem w 2004 roku jako tester aplikacji mobilnych przeznaczonych na rynek masowy. Przez następne lata nabierałem doświadczenia w obszarach Testowania i Jakości Oprogramowania, koncentrując się głównie na telekomunikacji oraz systemach mobilnych. W czasie swojej kariery byłem zaangażowany w testowanie, zarządzanie procesami testowania, szkolenia, wsparcie techniczne , analizę wymagań, rekrutację oraz tworzenie i inspekcje dokumentacji technicznej.

Poza doświadczeniem mobilnym i telekomunikacyjnym, zaangażowany także byłem w obszary systemów bankowych i finansowych. Obecnie pracuję na stanowisku Lidera Zespołu QA.

Posiadam dwa certyfikaty ISTQB poziomu zaawansowanego: Techniczny Analityk Testów oraz Kierownik Testów.

Mieszkam i pracuję we Wrocławiu.

Krzysztof Chyłła

Testować, czy nie testować?



To ma być pytanie?

Testować, czy nie testować? To ma być pytanie? Oddech. Przywołuję się do porządku i, zanim dam upust swoim poglądom na tak zaczepnie postawioną sprawę, pozwolę sobie najpierw dopełnić konwenansów.

Witam Cię i pozdrawiam Szanowny Kolego. Dawno już do siebie nie pisaliśmy - tym bardziej cieszy mnie okazja wdania się z Tobą w dyskusję o lekkim zabarwieniu polemicznym. Sprawdźmy, przetestujmy wzajemnie swoją argumentację. Niech zwycięży najlepszy!

Lepiej nie testować? Wolne żarty! Pamiętasz, „bo zupa była za słona" [kampania-spoleczna.pl]? Pół żartem, pół-serio, już jeden test wyeliminowałby takie ryzyko. Przejdźmy do meritum.

Po pierwsze primo. Co psują? Jak psują? Soft do testów już przychodzi zepsuty. Gdyby nie Ci dzielni chłopcy i dziewczęta, w pocie czoła wytężający wzrok i umysł w poszukiwaniu bugów, nawet by nie było wiadomo, że jest – już było – zepsute. Testerzy i testerki są moralnymi zwycięzcami każdego udanego projektu. Nie o pochwały tu chodzi, a o najwyższą jakość „za przystępną cenę". Zaprawiony w bojach specjalista od testowania nie da sobie napluć w kaszę i skutecznie odbije piłeczkę winy (ang. blamestorming) wspierając się merytoryczną potęgą metryk, których, nota bene, nie byłoby bez testowania wcale!

Po drugie primo. Wpadnij do mnie to Ci udowo…, a zaraz, momencik… Nie mogę przecież ujawnić materiałów kadrowych. Podpowiem tylko, że to nie muszą być całe organizacje, a odpowiednio skomponowane i rozwijane zespoły. Jest dochodowy projekt – jest business case. „Errāre hūmānum est" [Seneka]czyli tłumacząc z łacińskiego na testerski w dialekcie Yoda: "programista omylny jest". Nawet ten doskonale przeszkolony.

Tak z innej beczki, to przeszkolony z TDD programista w zwinnym zespole interdyscyplinarnym jest – nie bójmy się tego powiedzieć – testerem! Przynajmniej na część etatu. Tak jak tester-automatyk jest programistą.

Oszczędzać na testowaniu? Proszę Cię. Spójrz na koszt związany z wypuszczeniem oprogramowania niepoddanego testowaniu. Zapłaciłbyś za Tetrisa, który nie zapisuje najwyższych wyników, mimo że cała rozgrywka jest miodna do bólu?

Po trzecie primo. Tak, zgadzam się (Czy ja to na prawdę napisałem?). Nie da się przetestować wszystkiego, tak jak nie da się ściągnąć Internetu na dyskietkę (Sorry Chuck, przyp. red.). Popatrz jednak na

miliony graczy komputerowych niemogących się już doczekać 4.7GB kolejnej części swojej ukochanej gry z cyfrowej dystrybucji. Reasumując, nie wszystko, a to, co istotne i potrzebne z biznesowego punktu widzenia. Na to też są kolorowe wykresy.

Po czwarte primo. O gustach się nie dyskutuje. Niektóre się leczy. Chętnie zapoznam się z badaniami rynku pt. „Zapotrzebowanie na oprogramowania niskiej jakości w Europie i na świecie".

Po piąte primo ultimo (ha! wreszcie mogłem to napisać). Na przekór prawu Murphiego pozwolę sobie sparafrazować innego klasyka: „Testujmy co się da!" [Budka Suflera] i cash-back!

Nie zapominaj, że testowanie to frajda. A może udręka? Ale to chyba materiał na kolejną polemikę?

AUTOR

Krzysztof Chytla



Kierownik, projektant i automatyk testów z wieloletnim doświadczeniem w domenie systemów wbudowanych. Zdobywał doświadczenie w dużych projektach o międzynarodowym zasięgu, zapewniając najwyższą jakość produktu. Tester z krwi i kości z ciekawością i wnikliwością analizujący prężnie rozwijający się świat nowych technologii. Absolwent Wydziału Elektroniki Politechniki Wrocławskiej. Trener i coach, pasjonat zdobywania i dzielenia się wiedzą.

W życiu prywatnym fan literatury fantastycznej, fantastyczno-naukowej i gier planszowych przy szklaneczce zacnej whisky - najlepszej przyjaciółce redaktora.

WROTQA

WROCŁAW TESTING AND QUALITY ASSURANCE GROUP
Wrocław – Miasto Spotkań, znane też jako Polska Dolina Krzemowa – za sprawą grupy WrotQA (czyli Wrocław Testing and Quality Assurance Group) stał się miejscem spotkań również dla Testerów Oprogramowania.

W środę, 12 lutego 2014, miało miejsce pierwsze spotkanie o wdzięcznej nazwie WrotQA #1. Przybyła na nie liczna (ponad 50 osób) grupa profesjonalistów z zakresu zapewniania jakości, a także pasjonaci tematyki.

To długo wyczekiwana inicjatywa. Gratulujemy wykonania pierwszego kroku i życzymy powodzenia.

Więcej informacji o WrotQA na wrotqa.org

Łukasz Smolarski

Współpraca z informatykami z Indii

Wprowadzenie

Codzienność w międzynarodowych przedsiębiorstwach coraz częściej zmusza nas do pracy w międzynarodowym zespole. Dzieje się to między innymi z powodu coraz większych projektów, dążenia do redukcji kosztów pracy, ale także dostępności osób. Często serwis IT musi świadczyć usługi 24 godziny na dobę i dzięki zespołom z całego świata staje się to możliwe i bardziej dostępne. Jak widać z trendów na rynku IT, znaczna część pracy jest outsourcowana do Indii, gdyż informatycy z tego kraju uważani są za tanich a specjalistów. Same Indie słyną z ogromnej liczby specjalistów w IT. Większość młodych osób w Indiach ma lub będzie miała styczność z informatyką. Dzięki temu rozwój usług IT w tym kraju z roku na rok jest coraz większy. Często doświadczenie pokazuje, że praca z informatykami z tego kraju czasami może wydawać się nieefektywna, jednak warto zrozumieć, dlaczego i sprawić, aby współpraca była jak najefektywniejsza i przyjemna. Ja mając wieloletnie doświadczenie w pracy z Hindusami, uważam, że może być to bardzo ciekawe doświadczenie wielokrotnie owocujące sukcesem.

KILKA FAKTÓW

- Powierzchnia – prawie 3,3 miliona km2 (10 razy większa od terytorium Polski). Powierzchnia Indii to 2,4% powierzchni lądowej Ziemi.
- Liczba ludności – 1,21 miliarda w 2011 roku, czyli 17,5% ludności świata mieszka w Indiach. To 2,4 razy więcej niż liczba mieszkańców wszystkich państw Unii Europejskiej razem wziętych.
- Gęstość zaludnienia - 382 osób / km2 (2011 rok) (dla porównania w Polsce gęstość zaludnienia wynosi 120 osób / km2)
- Sex ratio – 940 kobiet na 1000 mężczyzn (2011)
- PKB (według parytetu siły nabywczej, z 2010 roku) – 3 176 $ / osobę, co daje 127 miejsce na świecie.
- Waluta: rupia indyjska (INR). 100 INR = 6,35 PLN (20 lutego 2011 roku).
- Główne wyznania: hinduizm (ponad 80%), muzułmanie, chrześcijanie, sikhowie, buddyści, dźiniści.
- Języki urzędowe: hindi oraz dodatkowo: angielski [1].

Kilka faktów na temat rynku IT w Indiach

- Aktualnie rynek IT oraz BPO w Indiach w sektorze eksportu prognozuje się na 12-14 % wzrostu w kolejnym fiskalnym roku tj. 2014 i osiągnie ok. 84-87 miliardów USD zgodnie z doniesieniami firmy Nasscom [2].

- Kolejne źródło donosi, że do 2017 roku liczba developerów w Indii prześcignie liczbę deweloperów w całych Stanach Zjednoczonych. Praca w IT w Indiach daje szansę na godziwe warunki życia, rozwój osobisty a także szanse na podróżowanie po całym świecie [3].

W całych Indiach istnieje 29 języków, a nieoficjalnie mówi się nawet o kilkuset. Często akcent Hindusów związany jest po części z ich lokalnym językiem i zależnie od regionu znajdziemy inne dialekty, naleciałości oraz mnóstwo różnić nawet w języku angielskim. Oficjalnie używane języki to Indyjski oraz Angielski. Indie są jednym z najbardziej zaludnionych krajów na świecie.

Odwiedzając np. Kalkutę czy inne duże miasto praktycznie nie jest możliwe znalezienie niezaludnionego miejsca. Na każdym kroku, nawet spacerując, przedzieramy się przez spory tłum ludzi. Indie są głęboko zakorzenione w wierze. Prawie każdy posiada złoty pierścień oraz obrazki swojego Boga, które w ich wierze chronią przed atakiem diabła. Przeważnie każda rodzina mieszka w licznej grupie tzn. dziadkowie, rodzice, dzieci w jednym domu. Hindusi pracują raczej zadaniowo, niż w trybie godzinnym. Często spędzają w pracy kilkanaście godzin dziennie. W ich społeczeństwie bardzo przestrzegana jest hierarchia pracownik – szef. Oczekują, że szef będzie ich nagradzał, promował a ich praca będzie zauważona. Co do jedzenia, to Hindusi preferują dość pikantną kuchnię. Ulubioną przyprawą jest curry. Dodatkowo przysmakiem i dodatkiem do niemal każdego dania są placki z pszenicy - często z takimi dodatkami jak czosnek, masło, itp.

Współpraca z Hindusami

Nawiązanie współpracy z Hindusami jest dość łatwe zarówno dla branży IT, jak i całego rynku. Jeśli chcemy prowadzić biznes niezwiązany z IT, powinniśmy odwiedzić Indie. Na każdym kroku spotkamy sprzedawców, zainteresowanych współpracą. Często odbywa się to tak, ze kupujemy na własny rachunek pierwszą partię i wysyłamy do naszego kraju, po czym kolejne transporty załatwia i organizuje nam już nasz współpracownik z Indii. My dokonujemy transferu pieniędzy i całą resztą zajmują się już Hindusi. Jeśli chodzi o współprace w IT, to wystarczy zgłosić się z chęcią współpracy do firm outsourcingowych za pośrednictwem internetu. Jest ich naprawdę sporo. Oto przykłady: (największe w Indiach) HCL [4], WiPro [5]. Wystarczy wysłać zapotrzebowanie, a firmy w krótkim czasie podeślą nam CV konsultantów wraz ze stawkami. Po wybraniu kandydatów, możemy sami przeprowadzić rozmowę i sprawdzić kompetencję, po czym rozpocząć współpracę.

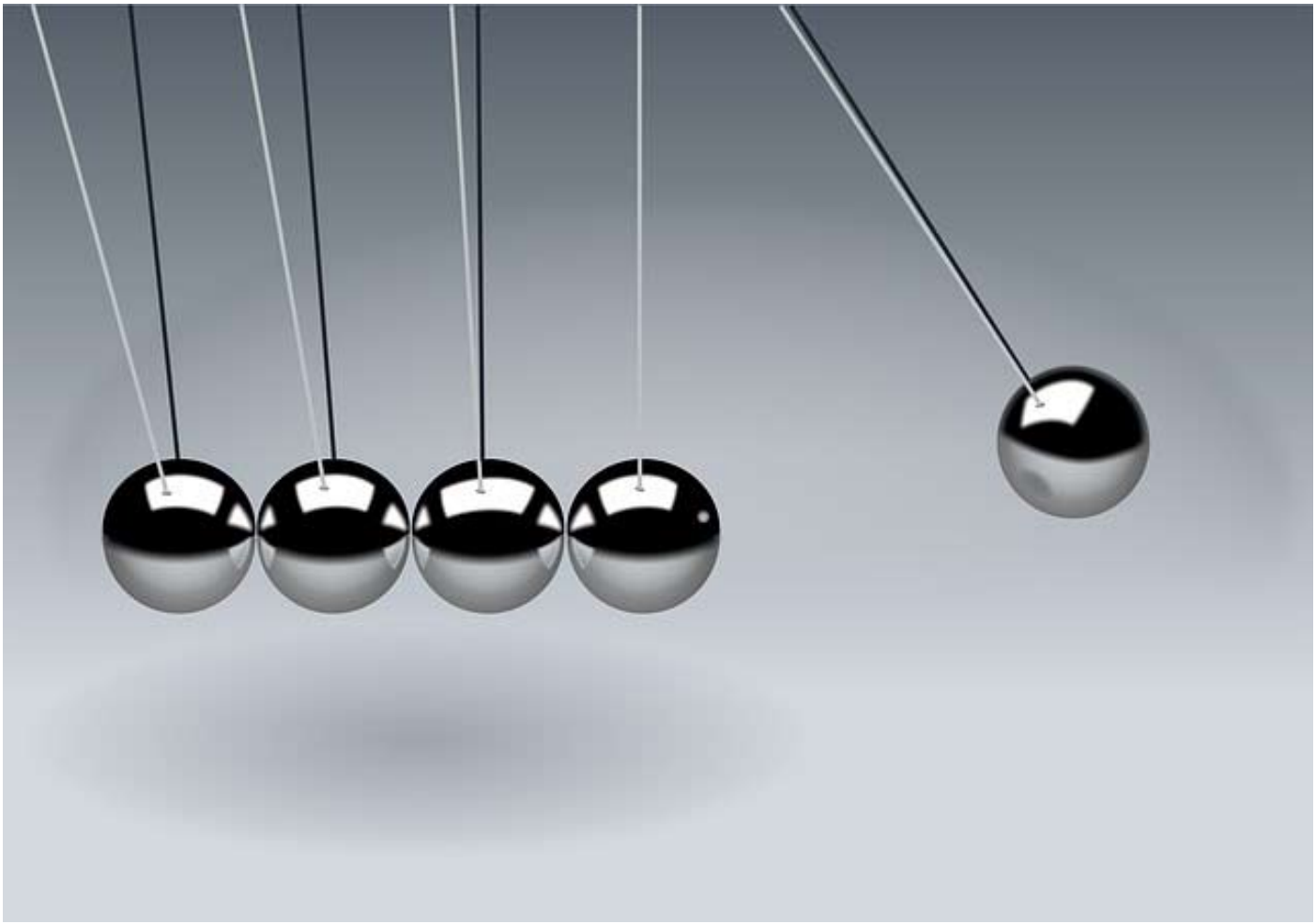
Aby efektywnie zacząć współpracę z ludźmi z Indii, należy już od samego początku przygotować się do rozmów. Od strony managera idealnie jest zorganizować spotkanie (może być konferencja przez telefon, wideokonferencja), gdzie każdy będzie mógł powiedzieć kilka słów o sobie. Często przydatna jest formuła zaczynająca zdanie „ Witamy na pokładzie nowych

specjalistów, którzy pomogą w tym, aby nasz projekt zakończył się sukcesem" lub coś podobnego, co zbuduje w Hindusach poczucie, że tworzą część czegoś ważnego a zarazem i odpowiadają także za swój kawałek pracy.

Kolejnym krokiem jest próba stworzenia obrazu pracodawca – przyjaciel. Chodzi o to, aby Hindusi mieli osoby w zespole za przyjaciół, a nie za tylko pracodawców żądających wyników pracy. Bardzo dobrze sprawdzającym się sposobem jest rozpoczynanie konwersacji od pytań o rodzinę, czas wolny pasję itp. Często podczas pracy z IT specjalistami z Indii, wymieniliśmy się zdjęciami, rozmawialiśmy po pracy przez portale społecznościowe i naprawdę po każdej współpracy zdobywałem nowych cennych przyjaciół, z którymi kontakty utrzymuję do dzisiaj. Dodatkowo w Indiach kultura pracy również jest nastawiona na przyjaźń. Idealne warunki do pracy to takie, gdzie panuje przyjacielska atmosfera.
Mozna spotkać opinie specjalistów z IT, że często ich praca z Hindusami nie układa sie efektywnie, nie przynosi rezultatów i wydaje się, że Hindusi nie wiedzą, co powinni robić. I właśnie tutaj jest przyczyna problemu tzn. najważniejsza we współpracy z Indiami jest komunikacja. Należy wszystko dokładnie i precyzyjnie ustalić, udokumentować i przekazać do zespołu w Indiach.

Poniżej przedstawiłem kilka kluczowych czynników, które mogą zadecydować o sukcesie współpracy:

- język - do komunikacji zawsze używaj prostego języka angielskiego i krótkich



zdań. Chodzi o to, abyś przekazał maksimum wiedzy w krótkich sentencjach używając prostego języka. Język komunikacji to przeważnie angielski. Unikaj skrótów związanych z projektem na początku współpracy, gdyż mogą zostać niezrozumiane i w efekcie przegapione.

- rozpoczynaj spotkania od zdań „How are you ?" .. lub podobnego i poświęć 1 minutę na spytanie, co słychać, jak minął dzień ? etc.

- rób notatki – podczas spotkania spisuj kluczowe decyzje a po spotkaniu staraj się wysłać podsumowanie zawierające jak najwięcej szczegółów dotyczących pracy. Spisz najważniejsze decyzje wraz z osobami przypisanymi do zadań, po czym roześlij takie streszczenie do wszystkich zainteresowanych.

- wyznacz lidera - jeśli w zespole jest więcej, niż jedna osoba z Indii, wyznacz osobę, która będzie liderem po stronie Indii i będziesz mógł w razie czego z nią się skontaktować i ustalić zadania oraz egzekwować postępy prac. Co więcej, wymagaj od lidera raportowania zadań i egzekwowania prac po stronie Hindusów.

- wyznaczaj sprecyzowane zadania – Unikaj pisania ogólnie o zadaniu lub problemie. Pracownicy z Indii lubią mieć jasno sprecyzowane zadania. Jeśli ma to być test aplikacji – opisz dokładnie, jaka część ma zostać przetestowana, podaj dane np. loginy oraz liczbę przypadków testowych i wskaż, jak właściwie wygląda praca testera w danym projekcie. Idealnie jest ustalic spotkanie i pokazać swój pulpit wskazując i opowiadając o konkretnych zadaniach.

- timeline/ deadline – Każde zlecone hinduskiemu zespołowi zadanie powinno mieć ustalone ramy czasowe. Jeśli zlecamy konkretne zadanie, to ustalmy najpierw na spotkaniu czas pracy i razem z zadaniem przekażmy wyraźnie, do kiedy każde zadanie ma zostać zrobione.

- przekazuj opinię do innych - Po skończonej pracy, jeśli została dobrze wykonana, nagradzaj. Nie chodzi tu o prezenty, lecz o zwykłego maila do innych, managerów pracowników z Indii i chwal

KILKA CIEKAWOSTEK

Wielokrotnie spotkałem prawdziwych przyjaciół podczas współpracy. Z początku wspólpracownik Hindus nie przesyłał na czas raportów. Prawie wogóle nie mieliśmy kontaktu. Po kilku próbach zadzwoniłem i postanowiłem poznac bardziej współpracownika. Podczas rozmowy oprócz strice technicznej rozmowy dowiedziałem sie ciekawych rzeczy jak np sytuacja, że dla mojego przyjaciela rodzice szukali żony, a on nawet nie widział na oczy przyszłej wybranki. Jak sie okazało mieszkał w Chennai, a tam tradycja jest taka, że rodzice wybierają kandydatkę, którą syn ma poślubić. Co więcej, mój kolega był szczęsliwy i wierzył w to, że rodzice na pewno wybiorą idealną kandydatkę na żonę, gdyż znają go bardzo dobrze. Jak się okazało ślub się odbył i do dzisiaj tworzą zgraną parę. Po ciekawej rozmowie, zauważyłem, że raporty zawsze były na czas a i współpraca okazała sie bardziej owocna.

ich pracę. To motywuje pracowników i na pewno kolejna współpraca z Tobą będzie owocna.

- unikaj stereotypów – to, że ktoś narzekał na jakość pracy ludzi z Indii nie znaczy, iż ma to spotkać i Ciebie. Unikaj wszelkiego rodzaju stereotypów i traktuj pracowników jako dobrych, chętnych do pracy, specjalistów. W dzisiejszych czasach jakość IT w Indiach bardzo się rozwinęła i bardzo często możemy spotkać wykwalifikowanych współpracowników z ogromną wiedzą i doświadczeniem.

- miej pozytywne nastawienie – Pamiętaj, abyś już od początku pozytywnie nastawiał się do pracowników. Pytaj o wakacje, czas wolny i wiedz, że są to osoby z innej kultury i mogą być między Wami róznice, jednak jesteście w tym samym projekcie i macie ten sam cel. Pokazuj swój optymizm i pozytywne nastawienie względem Hindusów.

Podsumowanie

Jak widać powyzej współpraca z hindusami może byc efektywna i owocna. Trzeba starać sie zrozumieć ich kulturę i być ich przyjacielem, a wtedy i praca staje się przyjemna i widać efekty. Podczas wielu lat poznałem mnóstwo znajomych, często z sobą rozmawiamy i są to naprawde wartościowi ludzie, którzy są swietnymi ekspertami IT.

REFERENCJE

[1] http://strefaindie.blox.pl/html/1310721,262146,21.html?851828
[2] http://www.ibef.org/industry/information-technology-india.aspx
[3] http://www.computerworld.pl/news/391122/Do.2017.roku.w.Indiach.bedzie.wiecej.deweloperow.niz.w.USA.html
[4] www.hcltech.com
[5] www.wipro.com

AUTOR



Łukasz Smolarski

Łukasz Smolarski jest absolwentem Wyższej Szkoły Biznesu-National Louis University w Nowym Sączu na kierunku Informatyka, Akademii Leona Koźmińskiego w Warszawie na kierunku Zarządzanie oraz Szkoły Głównej Handlowej – kierunek Project Management. Podczas studiów wygrał stypendium dla liderów ufundowane przez GE Foundation.

Swoją karierę zawodową rozpoczął na stanowisku Quality Software Engineer w firmie GTECH Poland, gdzie pełnił rolę Team Leadera oraz osoby odpowiedzialnej za automatyzację testów.

Obecnie pracuje w firmie HP Polska, jako Quality Manager biorąc udział w międzynarodowych projektach IT. Posiada certyfikaty ISTQB, PRINCE 2 a także w 2010 roku zdał certyfikat HP - Quality Center oraz QuickTestPro. Członek SJSI.

Kontakt: smolar2@op.pl

TestingCup 2014

II edycja Mistrzostw Polski w Testowaniu Oprogramowania TestingCup 2014

KIEDY i GDZIE

2-3 czerwca na Stadionie Narodowym w Warszawie odbędą się Mistrzostwa Polski w testowaniu oprogramowania.



Będzie to już druga próba wyselekcjonowania najlepszych testerów oprogramowania w Polsce. Pierwsze takie zmagania odbyły się we wrześniu 2013r. Wystartowało w nich ponad 100 testerów, którzy zmierzyli się z Mr Buggym – aplikacją stworzoną specjalnie na tą okazję. Tegoroczne dwudniowe Mistrzostwa mają przyciągnąć dwa razy większą ilość uczestników.

Mistrzostwa polegają na przetestowaniu aplikacji bazując na dostarczonej specyfikacji. Oprócz samej satysfakcji z wygranej na uczestników czekają także bardzo atrakcyjne nagrody, przewidziana pula ma wynosić 40 000 zł. Uczestniczyć mogą wszyscy, bez względu na doświadczenie w testowaniu, zarówno indywidualne osoby, chcące podjąć wyzwanie, jak i firmy, które chciałyby wydelegować swój zespół i przekonać się czy ich testerzy zasługują na miano Mistrzów.

Uczestnictwo w Mistrzostwach to jedyna w swoim rodzaju możliwość sprawdzenia umiejętności testerskich w praktyce. W odróżnieniu od dostępnych na rynku certyfikatów, tu weryfikuje się faktyczne umiejętności każdego z zawodników. Dodatkowo, każdy z nich otrzymuje certyfikat będący potwierdzeniem zdobytego wyniku.

To także okazja do zmierzenia się z nową, lepszą odsłoną Mr Buggy - aplikacji, która nie tylko zostanie przez Was przetestowana, a która po Mistrzostwach zostanie udostępniona za darmo.

Tegoroczne Mistrzostwa są równie atrakcyjne dla zawodników, jak i widzów. TestingCup to bowiem nie tylko zawody, ale też konferencja o tematyce testerskiej, nad którą opiekę merytoryczną sprawują testerzy.pl. Na Stadionie Narodowym przed uczestnikami wydarzenia wystąpi w sumie kilkunastu mówców, wśród nich Ci, których może już znacie, a także kilka nowych i ciekawych nazwisk, m.in. **Adam Roman** http://testingcup.pl/adam-roman/ i **Mateusz Olejarka** http://testingcup.pl/mateusz-olejarka/.

W agendzie znajdzie się też miejsce dla tajemniczego gościa, który z testowaniem będzie miał do czynienia po raz pierwszy w życiu podczas naszej imprezy, ale którego przygody, doświadczenia, miejsca jakie odwiedził(a) i rzeczy jakie przeżył(a) starczyłyby na pokaźną księgę.

Głównym prelegentem i gościem specjalnym Mistrzostw jest **Paul Gerrard**. Mało kto może pochwalić się takimi dokonaniami jak Paul. To jemu zawdzięczamy takie definicje w testowaniu jak faza testów, skalowalność, testowanie dostępności, testowanie oparte na ryzyku. W 2011 roku zdobył nagrodę EuroSTAR European Testing Excellence. W 2013 otrzymał The European Software Testing Awards (TESTA) za swój całościowy wkład w rozwój testowania. Będzie to jego pierwsza wizyta w Polsce, podczas której poprowadzi także warsztaty 4 i 5 czerwca, za których organizację odpowiadają testerzy.pl.

Drugim kluczowym prelegentem będzie **James Lyndsay** specjalizujący się w obszarze testowania eksploracyjnego. James jest częstym gościem na międzynarodowych konferencji oraz trenerem, prowadzi LEWT (London Exploratory Workshop in Testing), zaś jego prace zdobywają liczne nagrody.

Będzie też okazja do wymiany doświadczeń oraz uczestnictwa w panelach dyskusyjnych z prelegentami, partnerami oraz innymi uczestnikami Mistrzostw w specjalnie zaaranżowanej strefie integracyjnej.

To nie wszystkie atrakcje, które organizatorzy przygotowali z myślą o uczestnikach TestingCup 2014, zapraszamy do śledzenia strony wydarzenia http://testingcup.pl/ oraz ich profilu na facebooku: https://www.facebook.com/TestingCup.

TestingCup to jedno z najważniejszych wydarzeń polskiego świata testerskiego, nie może Was tam zabraknąć.

Zarejestruj się już dziś na: http://testingcup.pl/rejestracja/

Magazyn Quale objął tegoroczne Mistrzostwa patronatem medialnym.

Discover the different faces of Testing @ Belgium Testing Days

March 17-20, Bruges



Discover the Belgium Testing Days Conference – the innovative software testing "**doing conference**" that brings you hands-on workshops, interactive discussions sessions, advanced peer inter-communication, an innovative lab with the most up-to-date knowledge, tools, and technologies available in the industry today. The Belgium Testing Days is at its 5th edition. Meet **industry experts**, colleagues and peers in an international test and QA community for **4 days** jam-packed with **learning** & **doing sessions** that will help you make a powerful impact **in your job role** and for **your company**.

FACTS

1. 12 PRE-CONFERENCE WORKSHOPS
2. +58 SPEAKERS WITH INTERNATIONAL FAME
3. 5 KEYNOTES, INCL. 1 OPEN PANEL DISCUSSION
4. 5 ADVANCED WORKSHOPS
5. A INNOVATIVE LAB FOR DEV'ERS & TEST'ERS
6. TOPICS & CASE STUDIES COVERING OUR ACTUAL DILEMMA'S, ISSUES & CHALLENGES
7. INTENSE NETWORKING & INTERESTING SPONSORS

Visit us www.btdconf.com

Keynotes & Workshops by international famous knowledgeable colleagues


Dorothy Graham


Doug Hoffman


Fiona Charles


Stuart Read


Lisa Crispin


J.J. Cannegieter


Rob Sabourin

And Jerry E. Durant, Goranka Bjedov, Seretta Gamba, Isabel Evans, Derk -Jan De Grood & many more...

Natalia Ziajka

Dane testowe - własny pomysł czy narzędzie?

Kiedy warto generować dane oraz jakie generatory mogą nam pomóc?

Wprowadzenie

Jeśli jesteś testerem i piszesz przypadki testowe, to znaczy, że najprawdopodobniej masz potrzebę używania pewnych danych do przetestowania oprogramowania. Podczas testowania wpisujesz informacje w odpowiednie pola w celu przetestowania działania poszczególnych funkcjonalności w systemie. Te dane to właśnie dane testowe, które stosuje się w celu zbadania jakości oprogramowania.

Danymi mogą być zarówno dane wpisane ręcznie do aplikacji, jak i dane automatycznie odczytywane z plików (xls, xml , bazy danych , itp.) przez narzędzia automatyzacji.

Proces identyfikacji danych

Pierwszą czynnością w procesie identyfikacji zestawu danych potrzebnych w procesie testowania, jest określenie wymagań klienta dla danej aplikacji.

Dzięki temu można określić większość pozytywnych przypadków testowych i powiązanych z nimi danych.

Drugą czynnością jest sprawdzenie reguł dotyczących danych. Reguła sprawdzania poprawności ogranicza lub kontroluje dane, które użytkownicy mogą wprowadzać. Takich reguł poprawności danych może być wiele i mogą dotyczyć różnych warstw. Jeżeli weźmiemy jako przykład reguły sprawdzania poprawności wpisywanych informacji do pól tabeli, to wówczas wyodrębnimy warstwy:

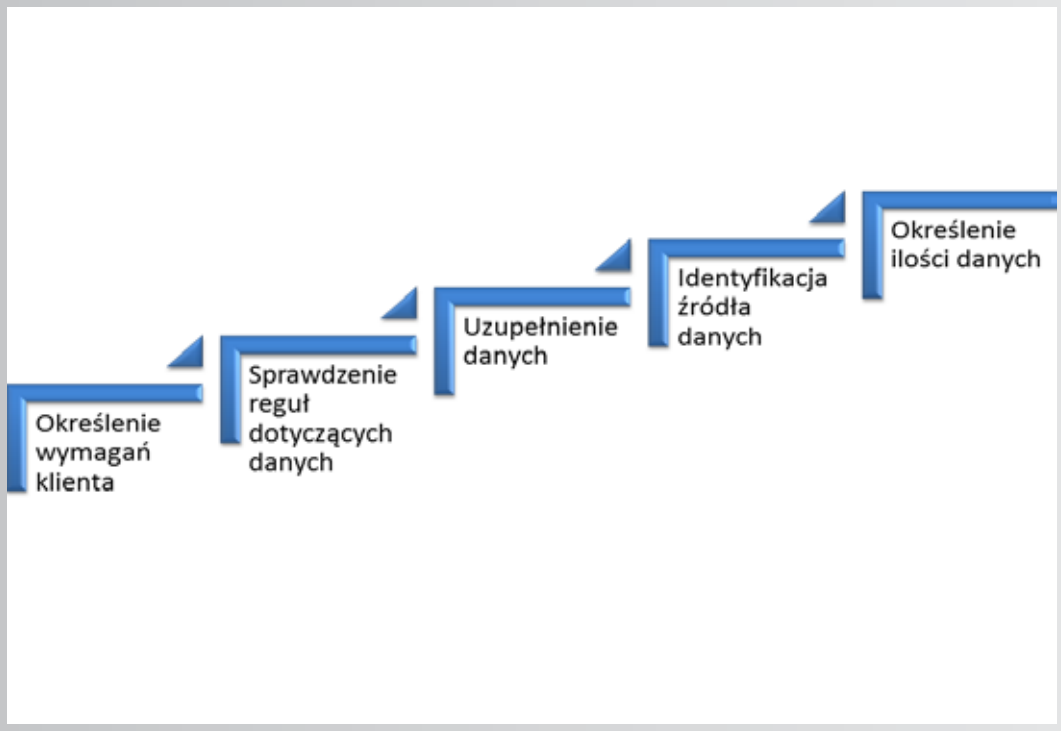
- Typy danych – to będzie nasza pierwsza warstwa poprawności danych.

Podczas procesu projektowania tabeli bazy danych definiuje się typ danych dla każdego pola w tabeli. Takie określenie pozwala ograniczyć możliwości wprowadzania danych przez użytkowników. Na przykład w polu typu Data/Godzina są akceptowane tylko daty i godziny; w polu typu Waluta są akceptowane tylko dane monetarne itd.

Ogólnie dane występujące w systemach informatycznych można podzielić na dane pozytywne i negatywne.

Pozytywne dane testowe to te, które powinny być akceptowane w systemach informatycznych. System w przypadku takich danych nie powinien zwrócić komunikatu błędu.

WAŻNE



Rys. 1. *Proces definiowania danych testowych*

Negatywne dane testowe są to dane, które nie powinny być akceptowane w systemach informatycznych. System po podaniu tych danych powinien zwrócić komunikat błędu.

Ze zbioru danych można wyselekcjonować dane optymalne, które skrócą czas testów. Wybrane dane optymalne potrafią wyszukać nawet do 80% najważniejszych błędów.

- Rozmiary pól - druga warstwa poprawności danych. Na przykład, pole do przechowywania imion można ustawić tak, aby było w nim akceptowanych maksymalnie 20 znaków. W ten sposób można zapobiec wklejaniu do tego pola dużych ilości niepotrzebnego tekstu przez niedoświadczonego lub złośliwego użytkownika.

- Właściwości tabeli - trzecia warstwa poprawności danych. Właściwości tabeli umożliwiają bardzo specyficzne sprawdzanie poprawności. Można na przykład ustawić, że dane pole jest obowiązkowe do wypełnienia i w ten sposób wymusić na użytkownikach wprowadzanie wartości w określonym polu.

Kolejną czynnością do wykonania w procesie identyfikacji danych, jest uzupełnienie zbioru danych o dane testowe.

Kiedy znane nam są dane testowe, możemy zająć się określeniem ich źródła pochodzenia.

Dane testowe mogą być generowane:

- Manualnie
- Poprzez skopiowanie danych produkcyjnych do środowiska testowego. Odwzorowują one wówczas rzeczywiste dane produkcyjne. W środowisku informatycznym możemy spotkać charakterystyczne aplikacje, w przypadku których znalezienie danych testowych może być stosunkowo trudne. Przykładem może być aplikacja, która wyświetla różne promocje na podstawie daty/czasu.

Jeśli chcielibyśmy używać danych produkcyjnych do testowania, to trzeba by było zmienić te dane na serwerze produkcyjnym, aby je przetestować. Byłoby to zachowanie aplikacji najbliższe rzeczywistemu działaniu funkcjonalności. Zmiana taka wpłynęłaby na wszystkich użytkowników tego serwera, dlatego należy tego unikać. Innym sposób to za pomocą zapytań SQL wyciągnąć dane z bazy danych i wykorzystać je do testów.

Przy tworzeniu zupełnie nowej funkcjonalności nie ma danych produkcyjnych, które można wykorzystać do testowania. Jest więc to sytuacja, kiedy przydatne jest generowanie danych [1].

- Poprzez skopiowanie danych z systemów od klienta

- Automatyczne wygenerowanie za pomocą generatorów danych. Mogą zostać wygenerowane zarówno dla przypadków pozytywnych, jak i negatywnych.

Ostatnia czynność to określenie minimalnej, maksymalnej i optymalnej ilość danych testowych.

Testowanie oparte na danych testowych

Pierwszą rzeczą, o której pomyślałam w związku z tym tematem, to testowanie w oparciu o przygotowane w Excelu dane. Jest to temat obecnie dla mnie bliski, gdyż wykonywałam takie testy w wersji podstawowej Soap UI, a obecnie robię to w wersji Pro. W mojej ocenie, praca z tym narzędziem bardzo upraszcza przygotowanie przypadków testowych.

W wersji darmowej narzędzia, aby pobrać dane z pliku, należy napisać skrypt w Groovy, natomiast w wersji Pro jest już do tego specjalnie przygotowany moduł, tak zwany Data- Driving- Testing.

Podam bardzo prosty przykład testu, dla którego ma sens przygotowanie takich danych w Excelu. Jednym z przypadków testowych było sprawdzenie, czy określeni użytkownicy mają dostęp do elementów technicznych z podanym id i krajem. Przygotowałam więc plik z id, danymi użytkowników (emaile) i kodem kraju. Dodatkowo, w ostatniej kolumnie pliku określiłam moje oczekiwane rezultaty w postaci 0 (nie ma dostępu) i 1 (ma dostęp).

Pobieranie danych w napisanym poniżej skrypcie polega na pobraniu za pomocą pętli kolejnych wpisów z pliku. Następnie następuje pobieranie lokalizacji z komórki Excela i ustawieniu właściwości lokalizacji w zmiennej, która jest przekazywana do zapytania Soapa.

Wynik otrzymałam w takiej postaci, jak poniżej (Rysunek 2). Jeden przypadek sfałszowałam i otrzymałam poprawny wynik - ostrzeżenie. Podałam informacje w pliku, że nie powinnam mieć dostępu, a tak naprawdę mam.

Przygotowanie danych testowych za pomocą generatorów

Podczas procesu testowego dla prostych aplikacji możemy nie potrzebować większej ilości danych testowych i możemy je szybko wymyśleć, jednak istnieją systemy, gdzie musimy sprawdzić działanie funkcjonalności na większej ilości danych lub sprawdzić wydajność aplikacji. Pojawia się wtedy problem z szybkim i poprawnym przygotowaniem danych – z tego powodu powstały generatory danych.

Narzędzia przygotowujące dane testowe manipulują danymi w bazie danych, za-

WAŻNE

```
//przechodzi przez kazdy wiersz w zakadce

while ( context.row < rowcount){

     //pobiera lokalizacje z komorki

     Cell SiteId = sheet.getCell( 0, context.row );
     String SiteIdValue = SiteId.getContents();
     Cell Email = sheet.getCell( 1, context.row );
     String EmailValue = Email.getContents();
     Cell Country = sheet.getCell( 2, context.row );
     String CountryValue = Country.getContents();

     Cell ExpectedResult = sheet.getCell( 3, context.row);
     String ExpectedResultValue = ExpectedResult.getContents();

// ustawia wlasciwosci location w zmiennej context, jest ona przekazywana do requesta:
testRunner.testCase.setPropertyValue("SiteId", SiteIdValue )
testRunner.testCase.setPropertyValue("Email" , EmailValue )
testRunner.testCase.setPropertyValue("Country", CountryValue )

//testRunner.testCase.setPropertyValue("exists", ExpectedResultValue )

if (ExpectedResultValue == "1") {
     testRunner.testCase.testSteps["CheckSiteExists"].assertions["exists"].disabled = false
     testRunner.testCase.testSteps["CheckSiteExists"].assertions["notexists"].disabled = true
} else if (ExpectedResultValue == "0") {
     testRunner.testCase.testSteps["CheckSiteExists"].assertions["exists"].disabled = true
     testRunner.testCase.testSteps["CheckSiteExists"].assertions["notexists"].disabled = false
}
log.info ("Testing for: " + SiteIdValue + ", " + EmailValue + ", " + CountryValue + " exists: "+ ExpectedResultVal
```

Rys. 2. *Przykład skryptu odpowiedzialnego za pobranie danych z Excela w Groovy*

```
Tue Feb 04 13:48:07 CET 2014:INFO:C:\Users\PLNAZIA\Documents\Web_services
Tue Feb 04 13:48:07 CET 2014:INFO:Testing for: 9AAS322836D4415, katarzyna.modzelewska@pl.abb.com , PL exists: 0
Tue Feb 04 13:48:07 CET 2014:INFO:OK
Tue Feb 04 13:48:07 CET 2014:INFO:Testing for: 9AAS058179O6850, natalia.ziajka@pl.abb.com, AE exists: 0
Tue Feb 04 13:48:08 CET 2014:WARN:FAILED
Tue Feb 04 13:48:08 CET 2014:INFO:Testing for: 9AAS806907A5812, natalia.ziajka@pl.abb.com, DE exists: 1
Tue Feb 04 13:48:08 CET 2014:INFO:OK
Tue Feb 04 13:48:08 CET 2014:INFO:Finished.
```

Rys. 3. *Przykład wyników testu z danymi pobranymi z źródła zewnętrznego - Excela*

wartością plików, czy transmisją danych, aby skonfigurować dane testowe i wykorzystać wszystkie możliwe ich kombinacje [2]. Takie dane są później często wykorzystywane do testów regresyjnych.

Generowania danych testowych jest jedynym niezawodnym sposobem, aby dokładnie wykonać testy wielokrotnie i konsekwentnie, wiedząc, że dane wejściowe się nie zmieniły. Zapoznając się z tematem generatorów, postanowiłam sprawdzić kilka dostępnych w internecie.

Generator http://www.testerzy.pl/wiesci-ze-swiata-testerow/generator

Testerzy.pl jako pierwsi w Polsce wyprodukowali Generator wspierający testowanie. Już ze wstępnych informacji dowiedziałam się, że jest możliwy do pobrania trial, którego pobrać i przetestować za darmo w ciągu 30 dni. Generator jest aplikacją desktopową przygotowaną dla systemu operacyjnego Windows i procesorów 32 bitowych.

Generator Danych Testowych pozwala na generowanie między innymi imion, nazwisk, adresów, e-maili, haseł alfanumerycznych i wielu innych przydatnych danych. Nadaje się do generowania danych do bazy danych SQL. Wyposażony jest także w mechanizm prezentowania wyników w formie HTML, pliku CSV oraz XML. Wśród wielu przydatnych danych można generować np. hasła, gdzie samodzielnie definiujemy, jakie znaki mają się pojawić na jakich miejscach hasła.

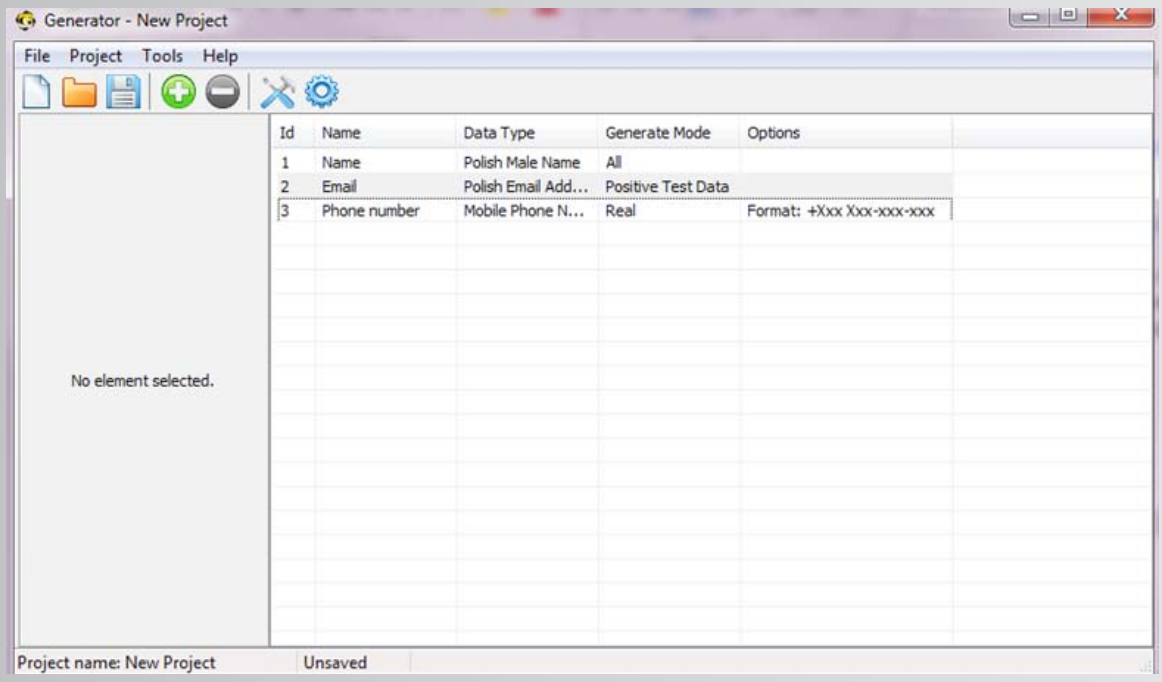
Mimo zalet jednak, natrafiłam na elementy, które zniechęciły mnie troszkę do tego narzędzia.

Nie ma możliwości zapisu danych do Excela, co wydaje mi się obecnie podstawą. Aplikacja uruchomiona w wersji angielskiej ma pomoc w języku polskim, ponadto nie jest przetłumaczona do końca poprawnie. Nie jest to zbyt intuicyjne narzędzie, więc pomoc powinna być napisana lepiej. W obecnej chwili pomoc w szerszej wersji znajduje się w książce [2].

Jeśli chcielibyśmy używać częściej tego narzędzia, to nie jest to możliwe, bo wersja darmowa jest na kilka dni, co w takiej sytuacji nie zachęca to użytkowania go już na samym początku.

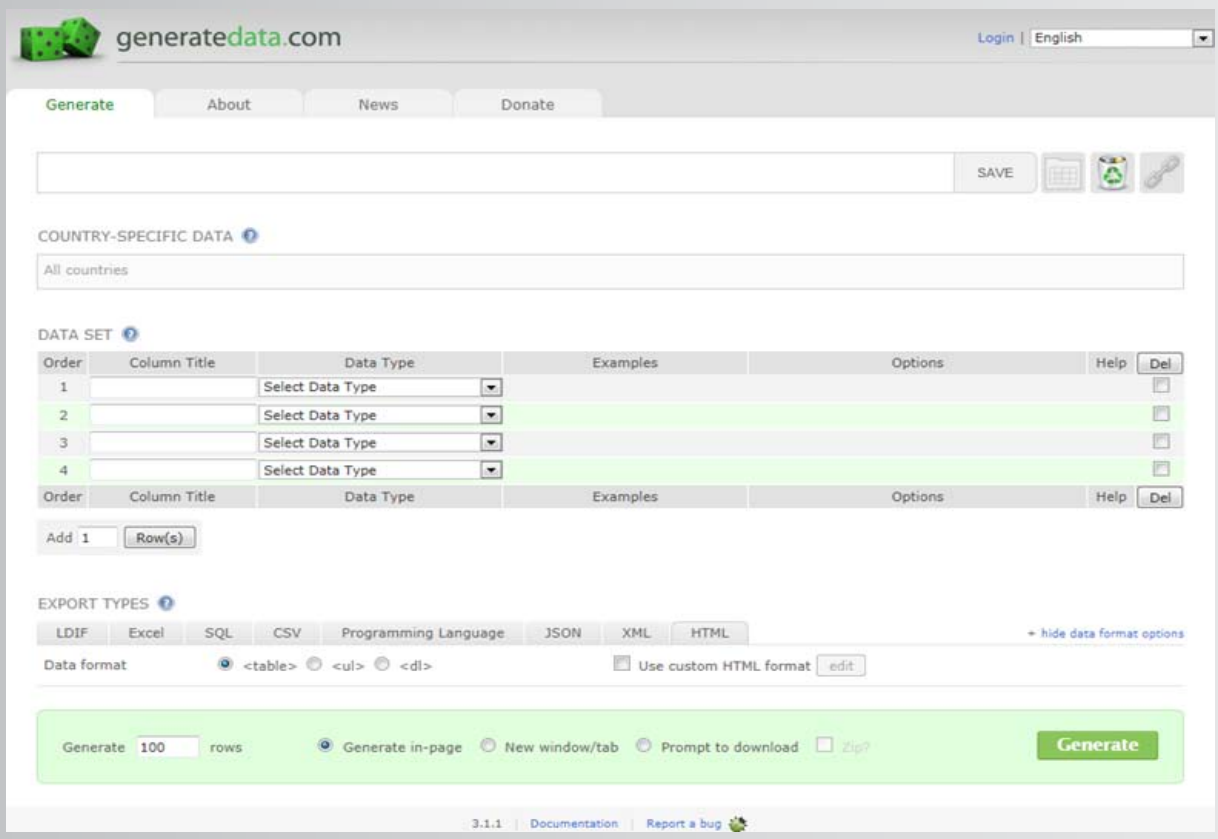
Generate data http://www.generatedata.com/

Generatedate.com oferuje darmową wersje internetową. Aby tworzyć własne projekty należy utworzyć konto i wykupić licencję lub pobrać skrypt i uruchomić go lokalnie. W wersji internetowej są już ograniczenia dotyczące ilości lokalizacji do wybrania, być może wersja rozszerzona zawiera wszystkie. Interfejs jest bardzo przyjazny użytkownikowi. Nowa wersja wydawana jest co miesiąc, więc mamy pewność, że produkt jest udoskonalany na bieżąco.

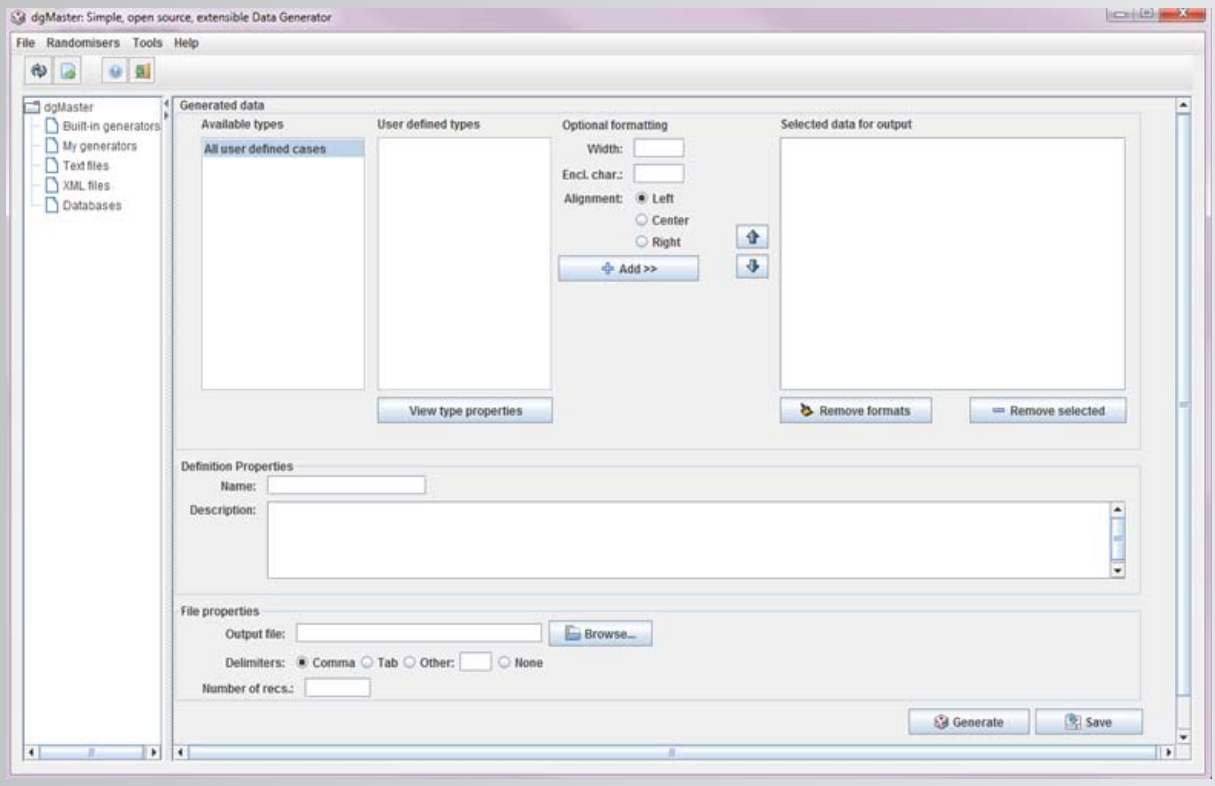


Rys. 4. *Interfejs programu Generator*

GENERATORY



Rys. 5. *Interfejs programu Generate Data*



Rys. 6. *Interfejs programu DG Master*

DgMaster
http://dgmaster.sourceforge.net/index.html

W tej chwili dgMaster jest zdolny do generowania danych tylko w formacie tekstowym (oczywiście, na liście uaktualnień znajduje się wsparcie dla bazy danych). Niemniej jednak, narzędzie oferuje typowe opcje, takie jak dostosowanie pola, wybieranie znaku separatora lub ustalenie stałych szerokości dla każdego elementu danych, itp.

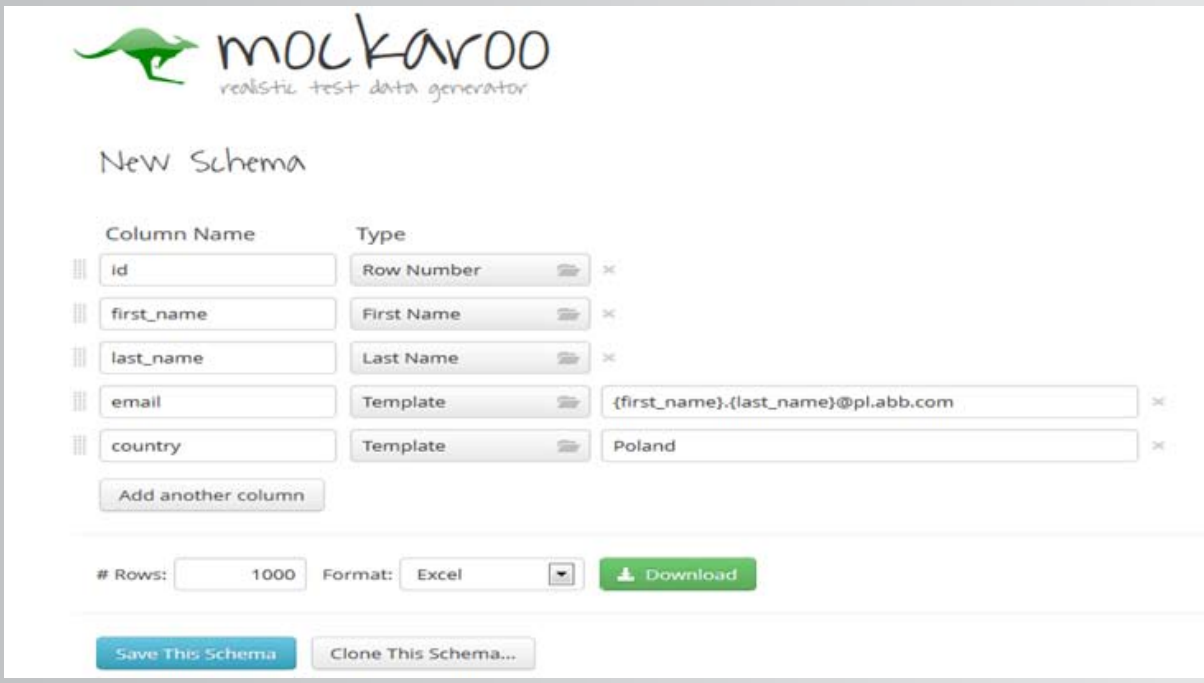
Niestety, już na początku pracy napotykam elementy, które jeszcze nie zostały zaimplementowane, jak podstawowa Pomoc, gdzie moglibyśmy zajrzeć, aby się dowiedzieć jak korzystać z tego narzędzia. Ostatnie poprawki zostały wprowadzone w 2009 roku, więc prawdopodobnie narzędzie to nie jest rozwijane od tego czasu.

Mockaroo
http://www.mockaroo.com/

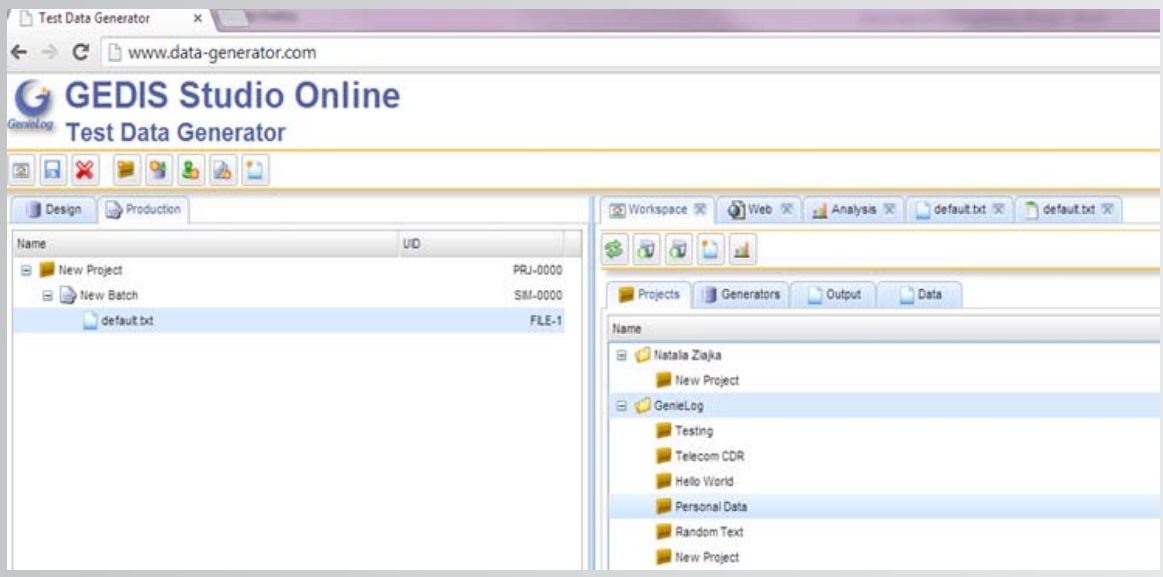
Mockaroo to narzędzie, które pozwala szybko i łatwo pobrać duże ilości losowo generowanych danych testowych w oparciu o własne dane techniczne, które można następnie załadować bezpośrednio do środowiska testowego. Mockaroo pozwala wygenerować do 100 tysięcy wierszy realistycznych danych testowych w formacie CSV, TXT, SQL, i JSON.

Nie jest wymagane programowanie.

Ten generator jest bardzo prosty i pozwala bez instalacji wygenerować dane, na przykład email z domeną Twojej firmy. Jeśli na dostępnej w narzędziu liście nie ma Twojej kraju, możesz swobodnie wpisać kraj, który chcesz, aby wystąpił w twoich danych. Mamy możliwość przygotowania swoich



Rys. 7. *Interfejs programu Mockaroo*



Rys. 8. *Interfejs programu GEDIS Studio Online*

schematów danych i ich zapisania, co jest plusem tego generatora. Jednak jest jednym z tych narzędzi, których dane pochodzą ze źródła zewnętrznego.

Gedis Studio http://www.gedis-studio.com/gedis-online.html

Gedis Studio - generator danych dostępny online. Umożliwia generowanie danych testowych bez konieczności instalowania żadnej aplikacji na komputerze.

Po ogólnym zapoznaniem się z generatorem, uważam, że jest warty zainteresowania.

Rejestracja zapewnia prywatną przestrzeń roboczą do przechowywania wszystkich plików danych zaangażowanych w projektach testowych na serwerach tej firmy. Posiada się również dostęp do wspólnych danych, takich jak dane referencyjne (wiele plików), bądź próbek projektów testowych.

Zaletą jest to, że aplikacja jest dostępna w każdej chwili z dowolnego miejsca i co więcej, można nawet poprosić o wspólną przestrzeń roboczą dla Twojej firmy do generowania danych. Można również podzielić się swoimi projektami z zarejestrowanymi użytkownikami.

TeDaGen http://www.tedagen.com/?page=data-generator

TeDaGen pozwala wygenerować nieograniczoną ilość realistycznych danych testowych w wielu formatach (plik tekstowy, XML, SQL, CSV, JSON, PHP, Ryby, Perl i Java Skrypt) bez potrzeby instalowania żadnego oprogramowania.

Narzędzie nie ogranicza się tylko do wparcia USA, ale obsługuje także inne kraje jak: DE, ES, FR, EN, NL. Generuje dane tylko za pomocą przeglądarki.

Jest to bezpłatne narzędzie.

Podsumowanie przeglądu

Podsumowując przegląd narzędzi okazało się, że występują zarówno wersje do zainstalowania lokalnie, ale też wersje inter-

netowe. Większość z nich posiada podstawowe funkcjonalności, do wygenerowania imienia, nazwiska czy adresu, ale są też takie, które są rozwinięte o inne funkcjonalności potrzebne do zarządzania danymi. Niemniej jednak znalezienie dobrego generatora wymaga czasu. Może się też zdarzyć, że jedynym sposobem na spełnienie konkretnych potrzeb danego projektu jest napisanie własnego generatora. Warto zainteresować się generatorami, ponieważ mogą nam one pomoc w codziennej pracy i zaoszczędzić czas, który możemy przeznaczyć na inne rzeczy.

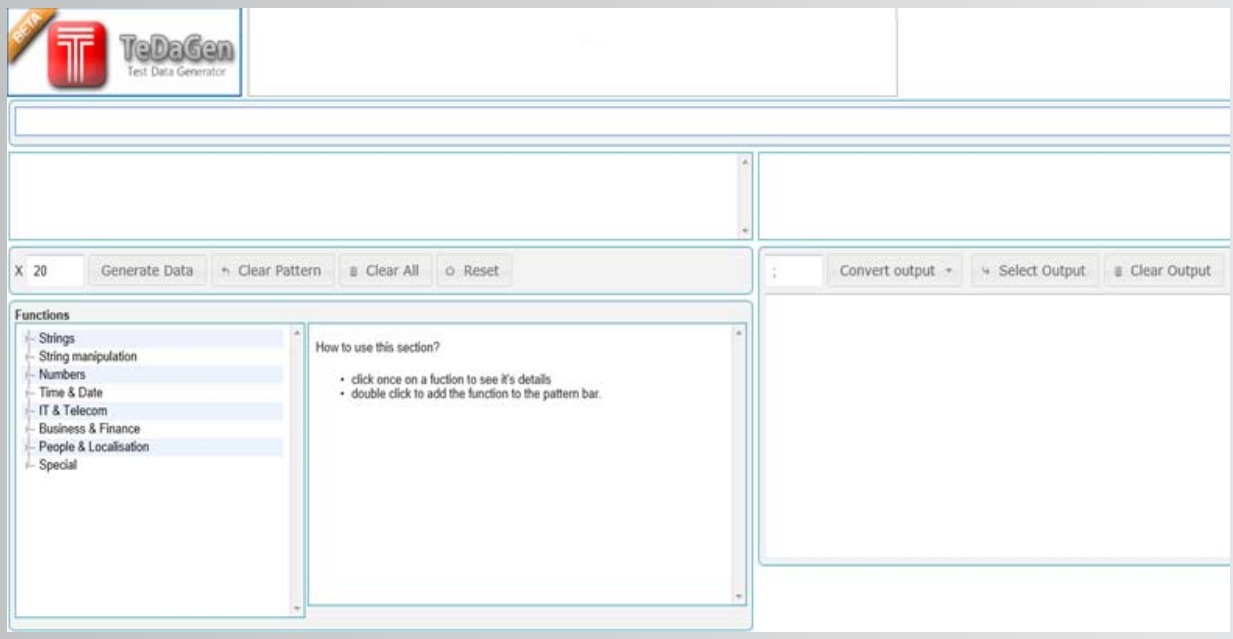
Zarządzanie danymi

Zbieranie danych tak naprawdę zaczyna się już od momentu przygotowania specyfikacji z wymaganiami (jakie dane mają być przyjmowane, przetwarzane i przechowywane) po testy akceptacyjne(sprawdzanie poprawności implementacji systemu przez wprowadzanie danych rzeczywistych).

Podczas procesu przygotowania danych powinniśmy pamiętać o ich gromadzeniu w repozytorium i wersjonowaniu, gdyż dane przygotowane w fazie wytwórczej mogą już nie być aktualne – i nie nadawać się do ponownego użycia.

Zalety centralnego przechowania danych:

- Rezygnacja z lokalnie przechowywania danych.
- Rezygnacja z kopiowania danych poprzez nośniki typu pamięć USB.
- Rezygnacja z każdorazowego ręcznego tworzenia danych.
- Zabezpieczenie danych przed niepowołanym dostępem.
- Możliwość ponownego użycia danych [2]



Rys. 9. *Interfejs programu TeDaGen*

Podsumowanie

Podsumowując narzędzia do testowania, mogę stwierdzić, że jeśli nie posiadamy szczególnych ograniczeń i potrzebujemy wygenerować przykładowe dane testowe, to możemy wykorzystać do tego przedstawione przeze mnie narzędzia. Część z nich udostępnia tylko podstawowe funkcjonalności do stworzenia danych, inne zaś zapewniają tworzenie całego projektu i zarządzanie danymi. Potrzeby każdego z nas są jednak różne i w pierwszej kolejności należy się zastanowić, czego potrzebujecie, jaki cel chcecie osiągnąć? Czasami wystarczy tylko przygotowanie kilka zapytań SQL, które pozwolą wam otrzymać potrzebne dane.

Odpowiednie dane podnoszą efektywność testów, poprzez ograniczenie czasu potrzebnego na testowanie i wykrywanie w skończonej jednostce czasu więcej defektów. Dobre dane podnoszą skuteczność testów w poszukiwaniu defektów dlatego, że badają obszary, w których człowiek popełnia najwięcej błędów. Mam tu na myśli zarówno programistów tworzących kod, jak i użytkowników, których akcje są w aplikacji walidowane.

W przeciwieństwie do przypadków testowych można z całą pewnością powiedzieć, że ilość „przydatnych" danych jest skończona. Pozwala to na podważenie tak popularnej wśród testerów maksymy, że testowanie nigdy się nie kończy.

Wyjaśniając to, każdy system informatyczny obsługuje pewny zakres danych (skończony), który jest zdefiniowany przez odpowiednie wymagania (biznesowe, funkcjonalne, niefunkcjonalne itd.).

Mając takie informacje, możemy stworzyć skończony zbiór danych testowych, w których znajdą się dane reprezentujące każdy możliwy „wzorzec" danych obsługiwanych przez system.

Dane testowe mogą być raz zdefiniowane i wielokrotnie używane w całym procesie tworzenia i testowania aplikacji. Co więcej, dane te mogą być również używane w pracach rozwojowych nad innymi aplikacjami. Należy tylko pamiętać, aby danymi testowymi zarządzać, tak jak każdym innym produktem prac, czyli kontrolować wersję i zabezpieczać przed przypadkowym usunięciem.

REFERENCJE

[1] http://watirmelon.com/2013/03/07/should-you-use-production-or-test-data-for-testing/ [dostęp 13 stycznia 2014r]

[2] Anna Piaskowy, Radosław Smilgin, Dane testowe. Teoria i praktyka. Wydawnictwo Helion, Gliwice 2011



Natalia Ziajka

AUTOR

Natalia Ziajka jest absolwentką Politechniki Krakowskiej na wydziale fizyki, matematyki i informatyki, na kierunku informatyka ze specjalnością grafika komputerowa. Od ponad 4 lat związana zawodowo z obszarem zapewniania jakości oprogramowania. Karierę w branży rozpoczynała jako tester i nadal udoskonala swoje umiejętności i powiększa doświadczenie w tej branży. Obecnie zatrudniona w Centrum Systemów Informatycznych szwajcarsko-szwedzkiej firmy ABB w Krakowie, znanej jako światowy lider w dziedzinie energetyki i automatyki.

Jej codzienne obowiązki związane są z opracowywaniem strategii testowania, przeprowadzaniem testów manualnych i automatycznych, monitorowaniem i optymalizacją procesu testowego, a także wdrażaniem nowych rozwiązań i standardów w zakresie kontroli i zapewnienia jakości.

ServIS, aplikacja, którą testuje, to globalne narzędzie do śledzenia i zarządzania produktami firmy i systemami ABB. System obsługuje wszystkie obszary biznesu i jest kluczową bazą danych w celu wspierania rozwoju działalności usługowej. ServIS jest światową aplikacją dostępną dla wszystkich pracowników ABB.

TEST WELL

6 marca 2014

Konferencja TEST WELL 2014
Więcej na: www.abb.pl/testwell

Power and productivity for a better world™ ABB

Excited by Quality?

III OGÓLNOPOLSKA KONFERENCJA DOTYCZĄCA JAKOŚCI OPROGRAMOWANIA

QUALITY EXCITES

Quality Excites jest konferencją ekspertów, której celem jest wymiana doświadczeń związanych z wdrażaniem idei tworzenia oprogramowania wysokiej jakości.

Tegoroczna edycja konferencji odbędzie się 31 maja w Gliwicach.

LISTA PROPONOWANYCH TEMATÓW

- Jakość procesu wytwarzania oprogramowania
- Innowacyjne podejścia i strategie do zapewniania jakości w projekcie
- Automatyzacja testów – projektowanie, narzędzia, zaawansowane techniki, studia przypadków
- Zarządzanie środowiskami testowymi i produkcyjnymi
- Automatyzacja procesu testowania – ciągłe testowanie – ciągła integracja – ciągłe dostarczanie – ciągłe wdrażanie
- Testy niefunkcjonalne: bezpieczeństwo, wydajność, użyteczność, testy A/B
- Czysty kod – wzorce projektowe, architektura, zarządzanie długiem technicznym
- Techniki xDriven-Development
- Zbieranie wymagań i estymacja zadań
- Czynnik ludzki oraz psychologiczne aspekty pracy w zespole
- Rozwój i role w zespołach Agile
- Coaching oraz mentoring – wsparcie dla zespołu

CALL FOR PAPERS TRWA DO 31 MARCA!

CFP

Formularz zgłoszeniowy znajduje się na stronie konferencji:

www.qualityexcites.pl

Dla kogo jest Quality Excites?

Konferencja jest dla tych, którzy są przekonani, że jakość jest warunkiem doskonalenia procesów wytwarzania oprogramowania. Zapraszamy testerów, liderów technicznych, inżynierów jakości, programistów zaangażowanych w testowanie oprogramowania, menadżerów testów oraz kierowników projektów. Innymi słowy, wszystkich, którym jakość pracy inżynierów nie jest obojętna.

Quality Excites odbędzie się 31 maja w Gliwicach. W programie znajdą się warsztaty przygotowane przez specjalistów firmy Future Processing oraz wykłady. Udział w konferencji jest bezpłatny.

Po raz kolejny organizatorzy dają możliwość wystąpienia z własną prelekcją. Call for Papers trwa do 31 marca. Formularz zgłoszeniowy znajduje się na stronie konferencji: www.qualityexcites.pl

Kto organizuje Quality Excites?

ORGANIZATOR

Organizatorem konferencji jest firma Future Processing, stanowiąca zespół ponad 500 doświadczonych inżynierów, tworzących systemy informatyczne, dedykowane dla przemysłu, mediów elektronicznych, opieki medycznej, sektora finansowego oraz transportu.

Przedsiębiorstwo znajduje się w czołówce zestawienia Deloitte dla najszybciej rozwijających się spółek technologicznych Europy Środkowej „Technology Fast 50". Konkurs „European Business Awards" przyniósł firmie laury „National Champion".

Oprócz Quality Excites spółka Future Processing organizuje międzynarodowy maraton programistyczny Deadline24, konferencję WUD (World Usability Day) Silesia oraz konferencję Future Dev Day.

Future Processing

POMYSŁODAWCY O QUALITY EXCITES

Daniel Dec



Software Quality Engineer & Technical Consultant w Future Processing

„Dążenia w kierunku budowania zwartego i interdyscyplinarnego zespołu to już nie mit, ale otaczająca rzeczywistość. Kiedy cały zespół jest odpowiedzialny za jakość, szybciej możemy uczyć się od siebie nawzajem. Kolejne retrospekcje pomagają usprawniać proces, dobierać nowe narzędzia i przeprowadzać analizy przyczynowo-skutkowe. Technologie sprawiają, że życie staję się łatwiejsze, a my możemy szybciej tworzyć lepsze oprogramowanie. Jakość zaskakuje, my jednak wolimy, aby nas ekscytowała. Przybądź na Quality Excites!”



Łukasz Kempny

Project Leader & Software Quality Assurance Engineer w Future Processing

„Głównym celem Quality Excites jest szerzenie idei testowania oprogramowania, ale nie tylko. Oprócz zagadnień związanych bezpośrednio z Quality Assurance, chcemy promować praktyki i techniki programowania, a także efektywnego zarządzania projektami. Jest już tradycją, że głównym punktem programu Quality Excites są warsztaty tematyczne. Powodów, dla których warto przybyć do Gliwic jest wiele: bezpłatne wykłady i warsztaty, które poprowadzą specjaliści z Future Processing oraz prelegenci z całej Polski, wymiana wiedzy i doświadczeń, możliwość poznania osób z branży oraz zaczerpnięcia nowych inspiracji.”

Tomasz Olszewski

Testowanie eksploracyjne

SŁOWNIK

Eksploracja - badanie nieznanych dziedzin lub terenów.[wikipedia]

Testowanie (oprogramowania) - jeden z procesów zapewniający jakość oprogramowania poprzez jego weryfikację i walidację

Testowanie eksploracyjne - Nieformalna technika projektowania testów, w której tester projektuje testy w czasie, gdy są one wykonywane i [w której] wykorzystuje się informacje zdobyte podczas testowania do projektowania nowych i lepszych testów. [James Bach] [Słownik wyrażeń związanych z testowaniem, wersja 2.0]

Prolog

Z pracą testera nieodłącznie wiąże się sytuacja zmiany projektu, nad którym pracuje. Zmiana taka przeważnie ma bardzo dobry wpływ na pracę: pozwala uniknąć rutyny, zwiększa aktywność, pozwala podnieść własne kompetencje oraz sprawdzić się w nowym środowisku, a co najważniejsze powoduje, że praca testerska jest ciekawa i nawet po latach ciągle pozwala odczuć powiew świeżości.

Praca nad nowym, dobrze przygotowanym projektem pozwala na wczesne rozpoczęcie i solidne przygotowanie się do procesu testowania w trakcie całego cyklu wytwarzania oprogramowania. Prędzej czy później jednak pojawia się projekt, w którym cały ten misterny plan legnie w gruzach. Moment, w którym mimo świetnego przygotowania oraz zaplanowania wystarczającego czasu i zasobów na rzetelne przetestowanie tworzonej aplikacji następuje opóźnienie w fazie analizy lub produkcji, zostaje skrócony termin zakończenia cyklu wytwarzania i/lub redukcja zasobów ludzkich.

W tym momencie w formie małej dygresji warto wspomnieć kwestie oprogramowania krytycznego ze względów bezpieczeń-

stwa, w którym z powodu konieczność audytowalności lub spełnienia wymaganych norm i procedur pewnych czynności nie da się skrócić lub ominąć. Oprogramowanie takie to rzadkość, a kierownictwo projektu nalega na dotrzymanie terminów i zobowiązań. W takiej sytuacji ciągle pozostaje nam kilka opcji do wyboru, a jedną z nich są testy eksploracyjne.

Testy eksploracyjne

Najprostszą definicją testowania eksploracyjnego jest równoległe projektowanie testów i ich wykonywanie przy jednoczesnym uczeniu się systemu. Jest to przeciwieństwo testowania w oparciu o skrypty, rozumianego jako wykonywanie predefiniowanych procedur testowych zarówno w wersji manualnej, jak i automatycznej, których celem jest sprawdzenie zgodności oprogramowania z wyspecyfikowanymi wymaganiami. Testowanie takie jest ukierunkowane przez przygotowany test plan oraz zaprojektowaną specyfikację. Testy eksploracyjne nie są przygotowywane zawczasu oraz nie są wykonywane zgodnie z dokładnie opracowanym planem. Ich celem jest wyeksponowanie informacji pożądanych przez grupę interesariuszy, ocena jakości oprogramowania, przy czym zgodność ze specyfikacją jest tylko jednym z wielu czynników.

Istotną kwestią, która charakteryzuje testy eksploracyjne jest poszukiwanie niespodziewanego zachowania aplikacji, nieoczekiwanego przez grupę testerów przed rozpoczęciem procesu testowania. Podstawowe techniki testowania promowane w trakcie szkoleń ISTQB skupiają się na oczekiwanym zachowaniu oprogramowania (wspieranym przez istniejącą dokumentację). Skutkuje to pominięciem kwestii związanej z nieoczekiwanymi zachowaniami w testowanym projekcie, co może skutkować niewykryciem błędów mających znaczący wpływ na odbiór produktu.

Testy eksploracyjne nie polegają na bezmyślnym „klikaniu". Podobnie jak inne techniki testowania, testy eksploracyjne również rozpoczynają się zgodnie z planem. Nie zawsze musi to być udokumentowany plan, często znajduje się on w głowie testera, a to co rozróżnia „standardowe" od eksploracyjnej techniki testowania jest adaptowalność i możliwość reakcji na zmieniający się kontekst w trakcie pracy. Tester może rozważyć użycie różnorakich technik na danym module, lub w wypadku gdy dana cześć aplikacji bezbłędnie przechodzi kolejne testy, skupić się na innych klasterach w oparciu o oszacowane ryzyko.

Testowa zasada Pareto mówi: 80% błędów znajduje się w 20% testowanych modułów. Możliwość wykonania pojedynczego testu w celu sprawdzenia zachowania



aplikacji pozwala na punktową ocenę pracy systemu, którego wyniki w połączeniu z oceną ryzyka pozwalają na zdobywanie wiedzy o produkcie i podjęcie decyzji na temat dalszej ścieżki testów umożliwiającej projektowanie kompleksowych i zdywersyfikowanych testów. Tego typu informacje nie sa nigdy uzystkiwane podczas usystematyzowanego procesu testowania. W testach eksploracyjnych całość sprowadza się do pytania: „Co się stanie z aplikacją gdy…" zamiast pytania: „Czy aplikacja spełnia wymagania".

W kwestii pracy z dokumentacją z jednej strony w zależności od wymagań stawianych przez interesariuszy możemy dostarczyć cały wachlarz opcji – od minimalnych do kompleksowych materiałów, a z drugiej w wypadku nieposiadania specyfikacji możemy znaleźć własną wyrocznię stworzoną w oparciu o doświadczenia z dziesiątek poprzednich projektów lub porównywać zachowanie aplikacji z już istniejącym produktem.

Tester w testach eksploracyjnych

Ilu testerów w projekcie, tyle podejść do samego szeroko pojętego testowania eksploracyjnego. Szeroki zakres umiejętności i kompetencji, doświadczenia wyniesione z różnych projektów, indywidualne cechy charakteru, różnorodne podejścia do rozwiązywania problemów i unikalne style pracy. Każdy z testerów świadomie lub nie korzysta w trakcie codziennej pracy z testowania eksploracyjnego, warto jednak ciągle poszerzać swoje wiadomości oraz umiejętności i świadomie używać całego spektrum dostępnych technik. Istnieje pewien zakres atrybutów, które ułatwiają bycie dobrym testerem eksploracyjnym:

- systematyczność pracy połączona z „szóstym zmysłem" do wykrywania anomalii i nieścisłości
- adaptowalność do zaistniałej podczas testów sytuacji
- wykraczanie podczas testów poza utarte schematy
- umiejętność wyszukiwania i wykorzystywania wyroczni testowej
- umiejętność pogodzenia ograniczonej czasowo sesji testowania eksploracyjnego z rozszerzonymi przeszukiwaniami błędogennych modułów (podparte analizą ryzyka)
- wczucie się w rolę zarówno nowego użytkownika, jak i eksperta rozwijanego oprogramowania
- nawiązywanie dobrych relacji z ekspertami domenowymi i wykorzystywania ich wiedzy w trakcie testów
- wykorzystywanie podobnych lub konkurencyjnych rozwiązań i zdobywanie wiedzy o testowanym produkcie
- chęć pogłębiania swojej wiedzy i zainteresowanie testowanym obiektem

Testy eksploracyjne pomagają w poznawaniu i uczeniu się testowanego produktu. Tester, jako osoba łącząca świat analizy biznesowej oraz świat techniki reprezentowanej przez zespół deweloperów, posiadająca wiedzę z obu dziedzin, generalnie ma dużo informacji na temat testowanego produktu. Wraz ze wzrostem wiedzy technicznej i domenowej dotyczącej testowanego produktu osoby zajmujące się testowaniem powinny mieć na uwadze również dodatkowy aspekt testowania – czy aplikacja jest użyteczna dla nowych lub mniej doświadczonych w danej dziedzinie użytkowników. Te zagadnienia poruszane są jednak przez kluczowe dla wielu systemów testy użyteczności, które będą opisane innym razem.

REFERENCJE

[1] Exploratory testing (by James Whittaker)
[2] Agile Testing (by Lisa Crisping and Janet Gregory)
[3] wikipedia.org
[4] http://www.satisfice.com/articles/what_is_et.shtml
[5] http://www.satisfice.com/articles/et-article.pdf
[6] http://www.computerworld.pl/artykuly/311111/Testy.w.pospiechu.html

Epilog

W sytuacji niewystarczającej ilości zasobów lub czasu, mając ograniczone możliwości postępowania testy eksploracyjne w znaczącej liczbie sytuacji będzie efektownym i efektywnym podejściem do testowania wytwarzanego oprogramowania. W wyniku testów od pierwszych godzin pracy otrzymujemy użyteczne wyniki i znajdujemy więcej błędów niż jakiekolwiek innej technice testowania. W wypadku presji wywieranej przez kierownictwo na dotrzymanie terminu realizacji projektu, mając świadomość niewystarczającego czasu i zasobów na wykonanie zaplanowanych testów, zamiast zastanawiać się, które moduły testować, a które pozostawić nietestowane, lub spędzać nadgodziny w pracy, które prowadzą do pogłębiania zmęczenia i frustracji – często warto zastanowić się i zmienić podejście. Testy eksploracyjne podparte analizą ryzyka w projekcie często mogę być rozwiązaniem naszego problemu. Może się to wiązać z mniejszą dokumentacją oraz mniej systematycznym podejściem do testów, ale wcale nie znaczy, że produkt będzie gorzej przetestowany, a raport z testów z informacjami o wykrytych defektach i ryzyku produktowym mniej rzetelny.

AUTOR

Tomasz Olszewski



Tomasz Olszewski – test inżynier w firmie Luxoft , absolwent Politechniki Wrocławskiej, miłośnik testowania, metodologii zwinnych i zdobywania nowych doświadczeń.

W wolnym czasie biegacz długodystansowy, wielbiciel gór i fan gwiezdnych wojen.

Maciej Zaborowski

W poszukiwaniu źródła innowacyjnych testerów

Firmy tracą czas i pieniądze na wyszukiwanie i selekcję pracowników. Gdzie pozyskiwać innowacyjnych testerów? Jak do nich dotrzeć? Na te pytania powinny odpowiedzieć działy personalne, które w ostatnim czasie poszukiwały nowych specjalistów do zespołów QC oraz QA.

Wrzesień 2013, miesiąc w którym odbyły się w Polsce pierwsze Mistrzostwa Testowania Oprogramowania. Chyba nikt nie spodziewał się aż tak dużego zainteresowania. Zapisy na Testing Cup zostały zamknięte pierwszego dnia rejestracji, a w przeciągu godziny było już zadeklarowanych ponad stu testerów. To kolejne tak duże wydarzenie jak w przypadku znanej przez wielu testerów konferencji Test Warez. Poza tym to pierwsza tego typu inicjatywa, gdzie testerzy oprócz wymiany doświadczeń mogli również rywalizować między sobą. Jak podaje organizator, jednym z celów Testing Cup jest promowanie testowania oprogramowania w Polsce. Jest to dość istotne patrząc na polski rynek IT, a zwłaszcza wspomniany obszar. Co ciekawe, według magazynu Forebs, Tester Oprogramowania znajduje się w pierwszej trójce zawodów, na które będzie największy popyt w 2014 roku [1].

Wracając jednak do głównych założeń Mistrzostw Testowania Oprogramowania, w przedstawionych celach [2] brakuje jeszcze jednego kluczowego argumentu, a mianowicie „podnoszenia i poszerzania kwalifikacji zawodowych". W pierwszej kolejności zwróćmy uwagę na aktualne sposoby rozwoju umiejętności testerskich. Najbardziej popularną metodą pozyskiwania wiedzy jest uczestnictwo w szkoleniach i zdobywanie certyfikatów. Nie zawsze jednak dyplom jest adekwatny do posiadanej wiedzy i zdobytych umiejętności. Kolejną z metod jest samodoskonalenie się poprzez edukację z dostępnych w sieci tzw. „samouczków". W Internecie możemy znaleźć odpowiedzi jak również gotowe rozwiązania i instrukcje na najczęściej pojawiające się pytania i problemy występujące w procesie testowania oprogramowania. Oprócz tego aby być na bieżąco z najnowszymi trendami, praktykami stosowanymi przez innych testerów warto jest czytać i śledzić magazyny branżowe. Treści zawarte w artykułach bardzo często stanowią istotny wkład uzupełnia-

jący naszą wiedzę. Dodatkowym źródłem informacji są także książki, natomiast ilość dostępnych publikacji jest ogromna i nie zawsze wiadomo co wybrać i od czego należy zacząć. Przeszukując serwis Amazon po frazie „testing" możemy znaleźć ponad półtora tysiąca pozycji! [3] Dlatego też, przed doborem odpowiednich lektur, warto jest zapoznać się z dostępnymi opiniami i recenzjami. Aktywny udział w wielu konferencjach testerskich jest również sposobem na poszerzenie swoich wiadomości jak również przełamanie barier związanych z autoprezentacją. Udzielanie się w społeczności testerskiej jest kolejnym krokiem do pozyskiwania wiedzy z zakresu testingu. Przykładem tego może być uczestnictwo w Mistrzostwach Testowania Oprogramowania.

Generalnie Testing Cup należy traktować jako kolejny projekt w karierze zawodowej testera. Jest to szansa, w której mamy możliwość wykazania się. Nie chodzi tu tylko o znalezienie jak największej ilości błędów oraz sporządzenie jakościowych raportów, ale przede wszystkim o pokazanie się jako ludzi, którym chce się rozwijać i robić coś więcej niż codzienne ośmiogodzinne „wypalanie się" w biurze. Życie testera to ciągłe doskonalenie swoich umiejętności, poznawanie nowych technologii, narzędzi, procesów biznesowych, jak również zdobywanie wiedzy z różnych sektorów gospodarki narodowej. Należy pamiętać, że wartościowi ludzie to tacy, którzy biorą odpowiedzialność za to co robią, którzy mają pomysł na siebie i umieją go zrealizować. Takim właśnie podejściem powinien kierować się dobry tester, co więcej strategia ta może również przynieść korzyści firmie, w której pracuje. Testing Cup to wydarzenie skierowane do obecnych testerów jak i do przyszłych specjalistów ds. kontroli i zapewniania jakości oprogramowania [4]. Wydarzenie w którym, ludzie mogą dowiedzieć się i zobaczyć czym jest testowanie oprogramowania. Impreza ta daje możliwość na swobodną wymianę doświadczeń i nawiązanie nowych kontaktów.

Pierwsza edycja mistrzostw zorganizowana była na bardzo wysokim poziomie, teraz czas na drugą edycję! Testing Cup 2014 odbędzie się już na początku czerwca. Tym razem impreza będzie trwała dwa dni, a w zawodach weźmie udział dwustu testerów. To znaczący postęp w porównaniu z ubiegłoroczną edycją, dlatego zapewnienie odpowiedniej promocji mistrzostw może zwiększyć rozpoznawalność tego wydarzenia na rynku IT. Mistrzostwa Testowania Oprogramowania umożliwiają praktyczną weryfikację umiejętności testerskich poszczególnych zawodników. Jest to korzystne zarówno dla firm rekrutujących testerów oprogramowania, jak również dla uczestników, którzy po zawodach otrzymują certyfikaty potwierdzające zdobyty wynik. W porównaniu z innymi konferencjami, czy też targami pracy, dzięki wstępnej preselekcji uczestników na Testing Cup pojawiają się nie tylko testerzy oprogramowania, ale przede wszystkim ludzie z doświadczeniem. W dodatku organizator imprezy zapewnia dostęp do strefy „networkingowej" dla sponsorów, co daje niepowtarzalną możliwość na bezpośrednią rekrutację specjalistów i zebranie CV od kandydatów. Bez wątpienia po kilku edycjach, Testing Cup może stać się znacznie bardziej rozpoznawalnym wydarzeniem i będzie zdecydowanie wyróżniać się na tle pozostałych inicjatyw.

Naprawdę warto, aby każdy tester choć raz miał możliwość uczestnictwa w Mistrzostwach Testowania Oprogramowania. Pamiętajmy, że Testing Cup to nie tylko zawody, ale również świetna zabawa, networking w kuluarach, czy też sposób na pozyskanie atrakcyjnej oferty pracy. Mr Buggy czeka na kolejnych pogromców błędów!



REFERENCJE

[1] http://kariera.forbes.pl/najlepsza-praca-w-2014-roku-10-najlepszych-zawodow-2014,artykuly,169016,1,1.html

[2] http://testingcup.pl/2013/?page_id=1019

[3] http://www.amazon.com/s/ref=sr_nr_n_0?rh=n%3A283155%2Cn%3A5%2Cn%3A4133%2Ck%3Atesting&keywords=testing&ie=UTF8&qid=1393092161&rnid=1000

[4] http://www.qualitygurus.com/download/QM001DifferenceBetweenQualityAssuranceAndQualityControl.pdf

AUTOR



Maciej Zaborowski

Maciej Zaborowski absolwent Wojskowej Akademii Technicznej, wydziału Cybernetyki na kierunku Informatyka. Certyfikowany tester ISTQB - Advanced Level - Test Manager.

Stał na czele zespołu, który zajął pierwsze miejsce w Mistrzostwach Testowania Oprogramowania.

Zawodowo, na co dzień zajmuje się kontrolą i zapewnianiem jakości w firmie świadczącej usługi doradcze. Posiada kilkuletnie doświadczenie w testowaniu małych jak również dużych/złożonych systemów. Uczestniczył w licznych projektach rekrutacyjnych, gdzie zajmował się wyszukiwaniem, selekcją i dostarczaniem konsultantów ds. testowania oprogramowania.

Prywatnie entuzjasta nowych technologii, otwarty na innych ludzi i nowe kontakty.

TestWarez 2014

Mobilis in mobili



CEL

Od lat cel konferencji pozostaje niezmieniony. Chcemy stworzyć sposobność do wymiany doświadczeń oraz zachęcić do dyskusji na temat problemów i trendów w testowaniu oprogramowania. Chcemy uczynić konferencję miejscem, w którym można pokazać swoje osiągnięcia i przedstawić problemy, z jakimi zmagają się organizacje wytwarzając i testując oprogramowanie.

Tematy konferencji koncentrują się przede wszystkim wokół narzędzi wspomagających testowanie oraz praktyki przygotowania i wykonania testów. Formuła konferencji pozwala na dialog z prelegentami i nawiązanie kontaktów w kuluarach. Nową koncepcją, dającą jeszcze większe możliwości do dyskusji pomiędzy uczestnikami konferencji, są tematyczne panele dyskusyjne.

Głównym tematem tegorocznej konferencji będą różne wyzwania związane z testowaniem na urządzeniach mobilnych i embedded. Zgodnie z postulatami uczestników konferencji Testwarez 2013, chcielibyśmy, żeby zostały poruszone również tematy związane z darmowymi narzędziami do automatyzacji testów oraz zarządzania testami, a także z testami bezpieczeństwa oraz różnymi rodzajami testów niefunkcjonalnych.

Kto organizuje

Konferencja jest organizowana przez Stowarzyszenie Jakości Systemów Informatycznych, które już od ponad dziesięciu lat stanowi platformę wymiany doświadczeń oraz miejsce spotkań i dialogu ludzi zajmujących się zapewnianiem jakości.

W roku 2014 zostanie zorganizowana dziewiąta edycja konferencji Testwarez.

Poprzednia edycja zgromadziła 250 uczestników.

HARMONOGRAM

- 1 maja 2014 –otwarcie rejestracji zgłoszeń na http://www.testwarez.pl/openconf
- 2 czerwca 2014 – zamknięcie rejestracji zgłoszeń

Do zgłoszenia należy dołączyć rozszerzony abstrakt (1800 – 4000 znaków) planowanego wystąpienia, zawierający opis podstawowych tez prezentowanego wystąpienia oraz prezentację zawierającą całą treść merytoryczną; ta wersja prezentacji nie powinna zawierać żadnych elementów pozwalających na identyfikację organizacji, z której prelegent pochodzi, ani samego prelegenta, gdyż będzie podlegała podwójnie ślepej ocenie.

- 27 czerwca 2014 – przekazanie wstępnych uwag do zgłoszonych prezentacji;
- 27 czerwca – 18 lipca 2014 – czas na ustosunkowanie się do uwag Komitetu Programowego przez autorów prezentacji;
- 1 sierpnia 2014 –akceptacja zgłoszonych propozycji wystąpień (Komitet Programowy);
- 29 sierpnia 2014 – termin nadesłania ostatecznych wersji prezentacji oraz artykułów.

Dla kogo

Konferencja jest skierowana do osób zaangażowanych na co dzień w problematykę testowania; zarówno do praktyków testów, kierowników testów, jak i do testerów początkujących.

Zapraszamy wszystkich, którzy chcą poznać sprawdzone praktyki, a także osoby bardziej doświadczone, które chciałyby poznać sposoby pracy i porozmawiać z podobnymi do siebie ekspertami. W tym roku szczególnie serdecznie zapraszamy do zaprezentowania swoich wyników:

- Testerów testujących aplikacje na urządzenia mobilne i embedded
- Testerów i kierowników testów pracujących z narzędziami darmowymi
- Testerów i kierowników testów wykonujących testy bezpieczeństwa oraz testy niefunkcjonalne
- Osoby chcące zaprezentować coś nowego – własne podejścia, best practices, ciekawe doświadczenia etc.

Zapraszamy też wszystkich innych, chcących podzielić się swoimi doświadczeniami w testowaniu.

Kiedy i gdzie

Koniec września, dokładne miejsce konferencji zostanie podane w terminie późniejszym.

Założenia programowe

Konferencja ma umożliwić przekazanie wiedzy, doświadczeń, informacji pomocnych w testowaniu i zapewnianiu jakości, gwarantujących uzyskanie i utrzymanie wysokiej jakości aplikacji i systemów IT. Możliwe jest przedstawianie wypracowanych przez siebie lub swoją organizację dobrych praktyk lub narzędzi wspomagających testowanie.

Podstawowym założeniem konferencji jest – jak w poprzednich edycjach – jej mocny wymiar praktyczny i narzędziowy.

TEMATYKA

PREFEROWANA TEMATYKA w podziale na zagadnienia:

- Projektowanie i wykonanie testów na urządzenia mobilne
- Narzędzia wspierające testy na urządzenia mobilne
- Darmowe narzędzia do automatyzacji testów
- Narzędzia do zarządzania testami
- Testy bezpieczeństwa, w tym penetracyjne
- Inne testy niefunkcjonalne

KOMITET

KOMITET PROGRAMOWY

- Karolina Zmitrowicz – CRM S.A.
- Jan Sabak – AmberTeam Testing Sp. z o.o.
- Joanna Kazun – SJSI
- Radosław Smilgin – testerzy.pl
- Sebastian Małyska – Genesys Telecommunication Laboratories

KOMITET ORGANIZACYJNY

- Lucjan Stapp – Politechnika Warszawska
- Maciej Chmielarz – Asseco Poland SA
- Bartłomiej Prędki – Credit Suisse
- Sławomir Abramczyk – SMT Software

Forma wystąpień

Prezentacje standardowo mają trwać 45 minut (w tym czas na Q&A), przy czym przy każdej sesji przewiduje się dodatkowy czas na podsumowanie. Prezentacja nie powinna przekraczać 15 slajdów. Do prezentacji można załączyć dodatkowe materiały uzupełniające, do włączenia w elektroniczne materiały konferencyjne oraz do umieszczenia na stronach konferencji do pobrania. Prelegenci zobowiązują się do przygotowania artykułu opartego na swojej prelekcji, o objętości 4-10 stron w formacie A4. Artykuły zostaną opublikowane w formie materiałów pokonferencyjnych w magazynie tematycznym.

Uwagi dodatkowe

- Prelegenci uzyskują prawo do darmowego uczestnictwa w całej konferencji oraz pokrycie kosztów 1 noclegu prelegenta w pokoju dwuosobowym w wybranym przez organizatora hotelu.
- Przesłanie prac stanowi równoznaczne nadanie im licencji CC-BY (http://creativecommons.org/licenses/by/3.0/pl/)
- Wystąpienia będą nagrywane i zostaną udostępnione on-line po konferencji

Prośba do prelegentów

Bazując na doświadczeniach z poprzednich konferencji, oczekujemy, że większość uczestników będzie zaznajomiona z faktami podstawowymi, w szczególności w zakresie procesów wytwarzania i testowania oprogramowania. Dlatego prosimy, aby w swoich prezentacjach unikać wprowadzeń, wyjaśniających znaczenia takich podstawowych elementów jak wymagania, testowanie itd. Prowadzi to do niepotrzebnego przedłużenia prelekcji i znużenia słuchaczy już na samym początku prezentacj



Magazyn

Wydawca
VWT Polska Michał Kruszewski
Przy Lasku 8 lok. 52, 01-424 Warszawa
Numer NIP 5272137158
Numer REGON 142455963

Redaktor naczelny
Karolina Zmitrowicz
karolina.zmitrowicz@quale.pl

Zastępca redaktora naczelnego
Krzysztof Chytła
krzysztof.chytla@quale.pl

Redaktorzy
Bartłomiej Prędki
bartlomiej.predki@quale.pl
Michał Figarski
michal.figarski@quale.pl
Piotr Poznański
piotr.poznanski@quale.pl

WWW
www.qualemagazine.com
www.quale.pl

Facebook
http://www.facebook.com/qualemagazine

Reklama
info@quale.pl

Współpraca
Osoby zainteresowane współpracą w zakresie publikacji prosimy o kontakt:
info@quale.pl

Wszystkie znaki firmowe zawarte w piśmie są własnością odpowiednich firm.

Prawa autorskie
Wszystkie opublikowane artykuły są objęte prawem autorskim autora lub przedsiębiorstwa. Wykorzystywanie w innych publikacjach, kopiowanie lub modyfikowanie zawartości artykułu bez pisemnego upoważnienia autora jest zabronione.

Grafiki użyte w magazynie

Okładka

Architecture Blue Building Business City http://pixabay.com/en/architecture-blue-building-business-22231/

Pozostałe

Laser Laser Light Research Laboratory Science http://pixabay.com/en/laser-laser-light-research-11646/

Pen Office Leave Office Accessories Lid http://pixabay.com/en/pen-office-leave-office-accessories-62374/

Newton's Cradle Balls Sphere Action Reaction http://pixabay.com/en/newton-s-cradle-balls-sphere-action-256213/