PARASOFT®

Zalety systematycznego wprowadzania testów niefunkcjonalnych

TESTWAREZ 2013

2013-10-03

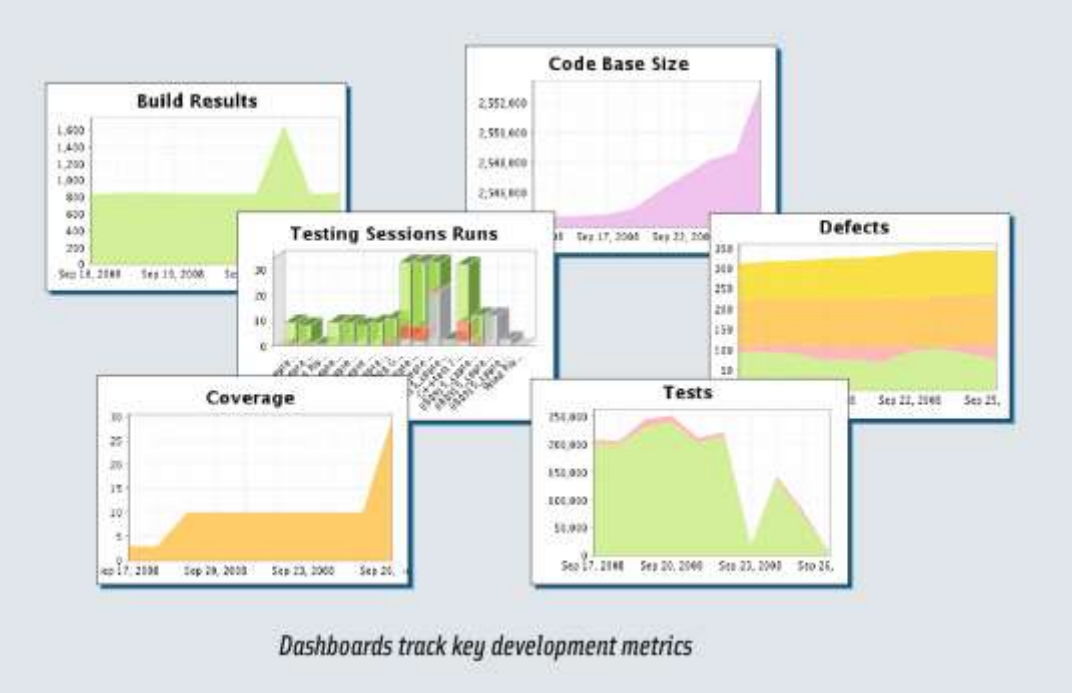
O Parasoft

Parasoft tworzy narzędzia, które wspierają efektywność całego procesu SDLC:

- Analiza statyczna , unit testy, etc.
- Agregacja danych i raporty z procesu SDLC
- Testy funkcjonalne
- „Inteligentne” symulatory

Parasoft a testy automatyczne

- Robimy testy automatyczne API
- Ponad 500 tys. testów dla jednego z projektów



Dashboards track key development metrics

Host Compilers

- Windows
  - Microsoft Visual
  - GNU and MingW gcc/g++
  - GNU gcc/g++
  - Green Hills MULTI for Windows
- Linux 32 and 64 bit processor
  - GNU gcc/g++
  - Green Hills MULTI for Linux
- Solaris
  - Sun ONE Studio
  - GNU gcc/g++
  - Green Hills MULTI for SPARC Solaris

Target/Cross Compilers

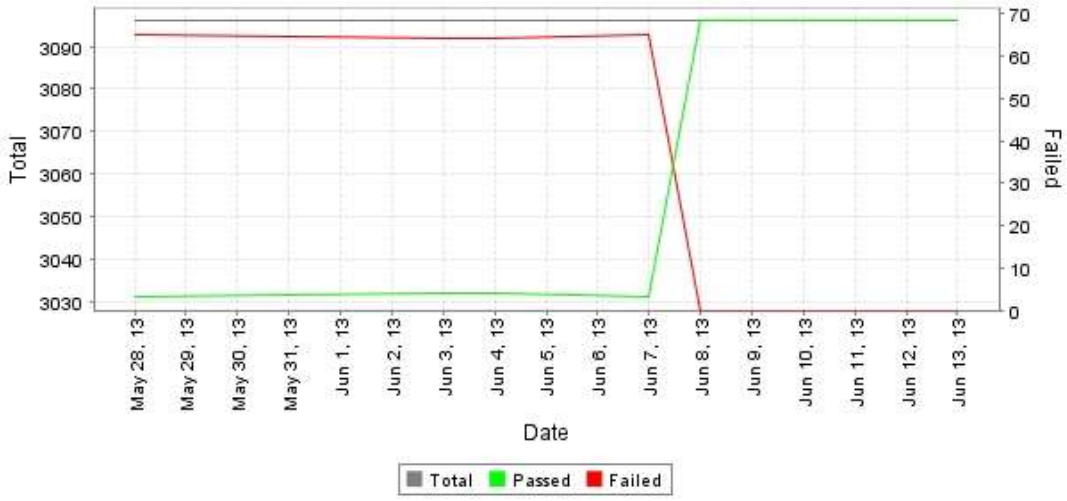
- Altera NIOS GCC
- ADS (ARM Development Suite)
- ARM for Keil uVision
- ARM RVCT
- ARM DS-5 GNU Compilation Tools
- Cosmic Software 68HC08
- eCosCentric GCC
- Freescale CodeWarrior C/C++ for HC12
- Fujitsu FR Family SOFTUNE
- GCC (GNU Compiler Collection)
- Green Hills MULTI for V800
- IAR C/C++ for ARM
- IAR C/C++ for MSP430
- Keil C51
- Microsoft Visual C++ for Windows Mobile
- Microsoft Embedded Visual C++
- QCC (QNX GCC)
- Renesas SH SERIES C/C++
- STMicroelectronics ST20
- STMicroelectronics ST40
- TASKING 80C196 C
- TASKING TriCore VX-toolset C/C++
- TI TMS320C2x/C2xx/C5x
- TI TMS320C2000 C/C++
- TI TMS320C54x C/C++
- TI TMS320C55x C/C++
- TI TMS320C6x C/C++
- TI MSP430 C/C++
- Wind River GCC
- Wind River DIAB

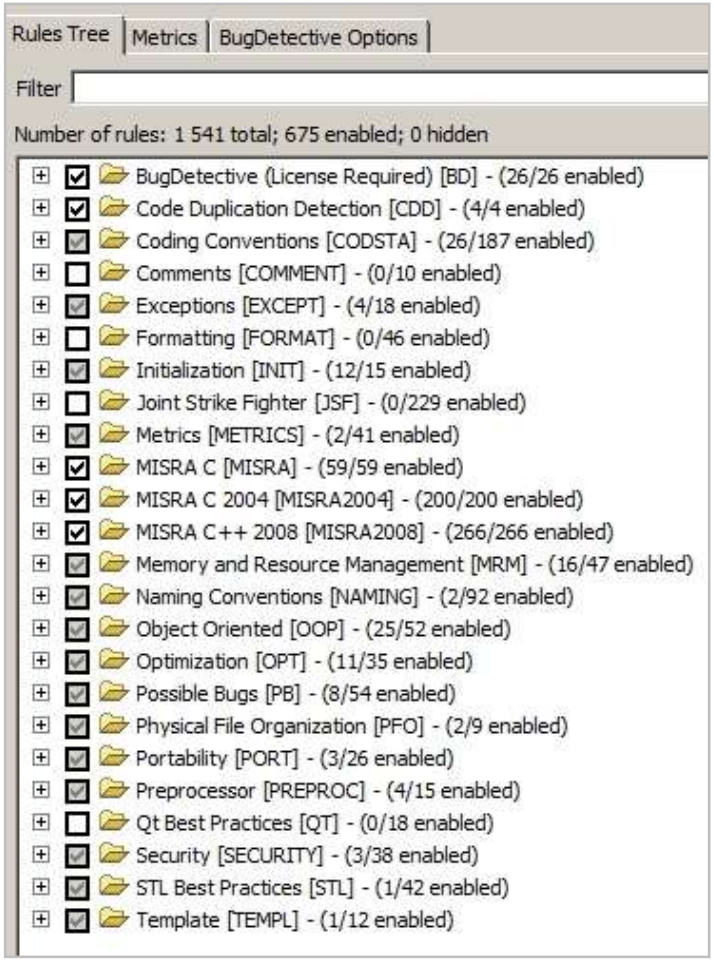
Automatyczne testy niefunkcjonalne

PARASOFT

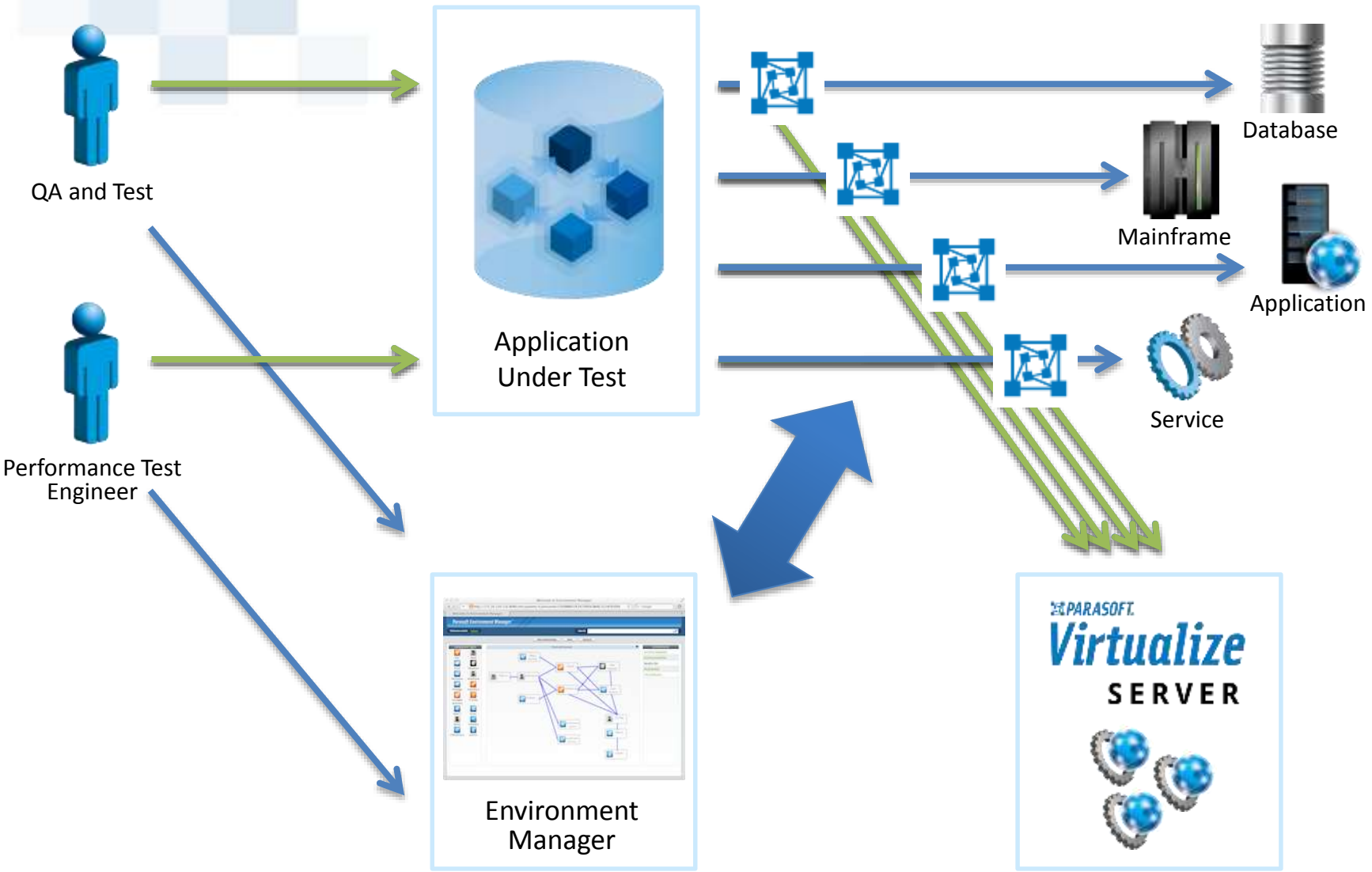
Mamy testy automatyczne dla :

- Analizy Statycznej
- Unit Testów





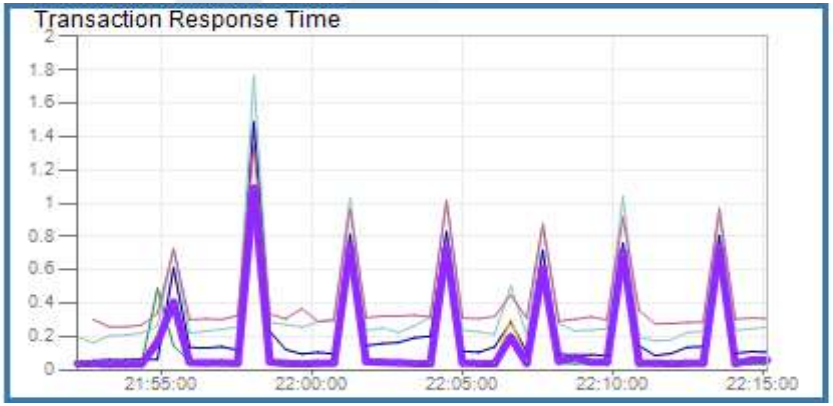
Nowy Produkt - Parasoft Virtualize

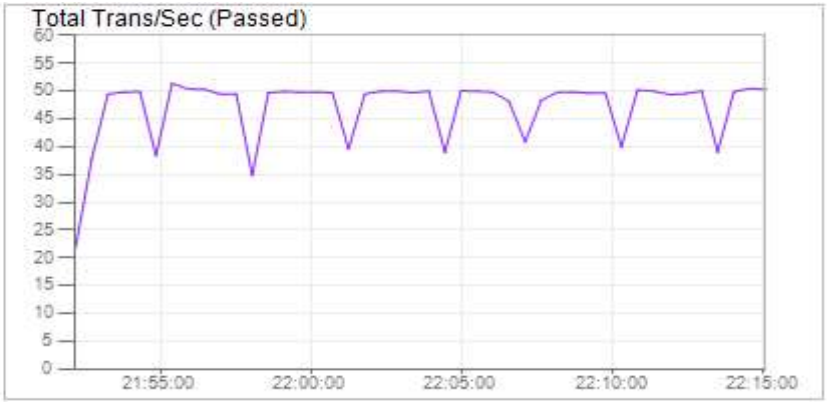


Przypadek klienta

- Duży kontrakt.
- Prestiżowy klient.
- Używa Parasoft Virtualize do symulacji zewnętrznych oraz wewnętrznych systemów, aby zredukować czas potrzebny do przeprowadzenia testów.

Pojawiają się problemy…



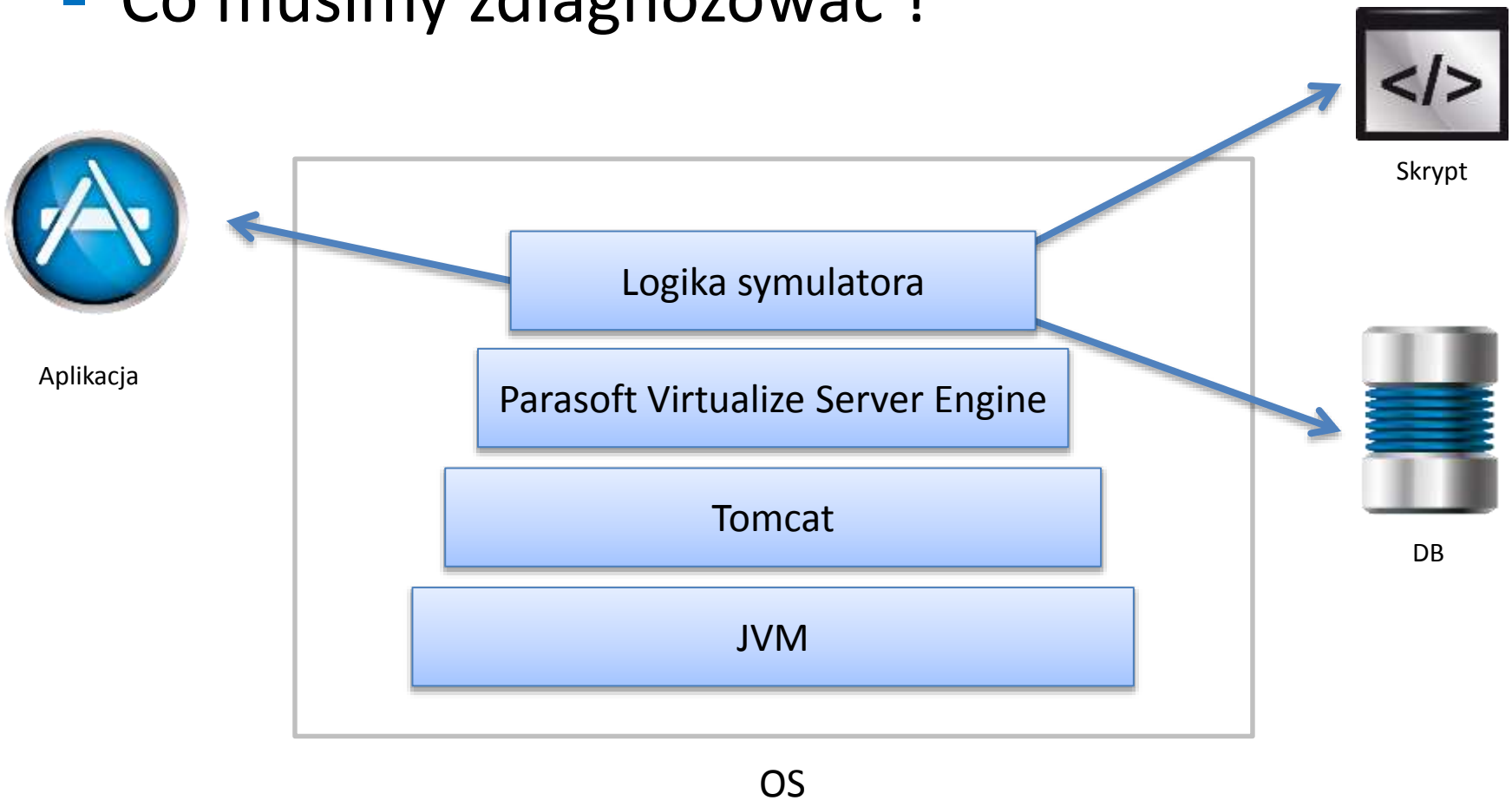


Co się dzieje u klienta ?

- Wsparcie techniczne usiłuje zdiagnozować przyczynę.
- Przesyłamy nowe instrukcje jak uruchomić środowisko.
- Klient w miarę szybko odpowiada – niestety nie mamy zauważalnej poprawy.
- Eskalujemy problem..

Gdzie może być problem…

- Co musimy zdiagnozować ?



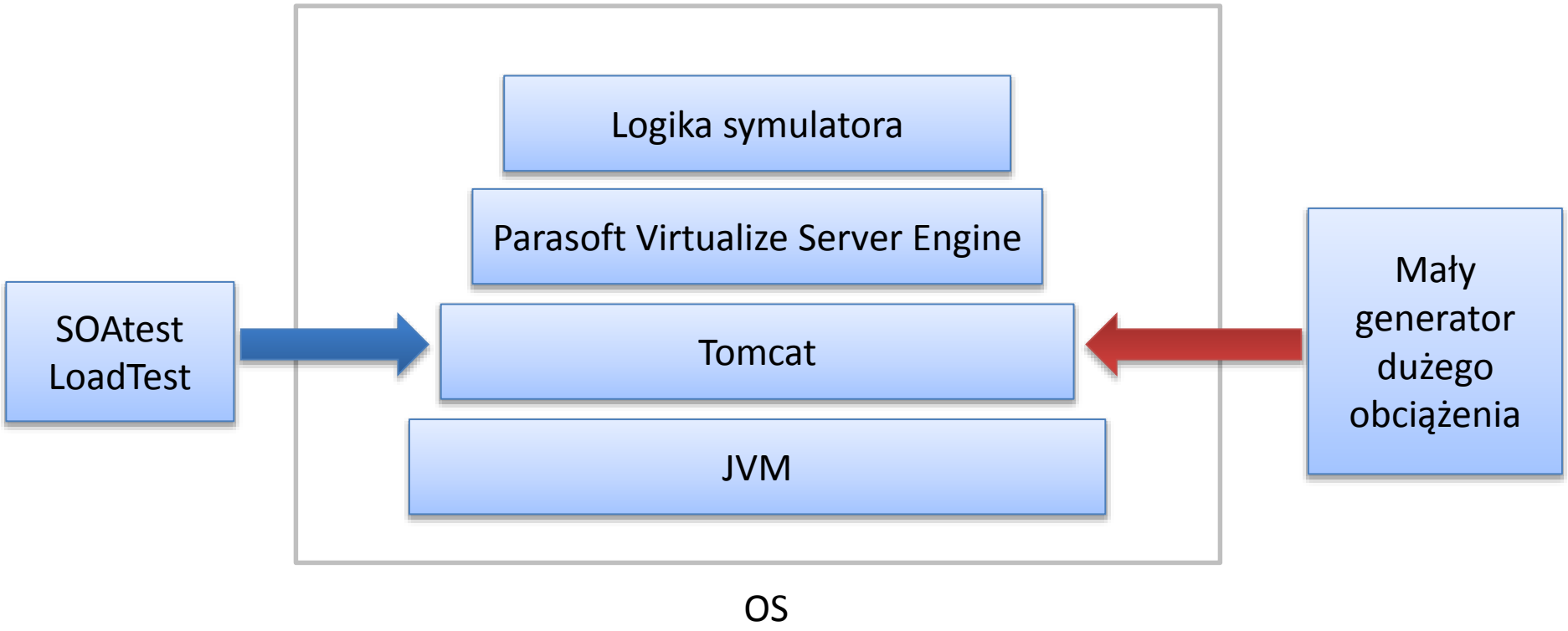
Debug-owanie – co musieliśmy sprawdzić

PARASOFT

- Web server i JVM configuration – Thread and Connection Pools, GC strategy, memory heap allocation, etc.
- Silnik Virtualize
- Wewnętrzy silnik odpowiedzialny za logikę symulatorów (PVA)

Jak to przetestowaliśmy?

- Nasze środowisko do testów



Parasoft Virtualize - testy

- Rodzaje testów:
  - Logika respondera: niska , średnie i wysoka high complexity
  - Używane źródła danych zawierają 10-100,000 wierszy
  - Odpowiedzi respondera mogą być tworzone przy pomocy testów
  - Dodatkowo mogą wystąpić dane które są hierarchiczne
  - Duży rozmiar odpowiedzi :od 100k do 10M

Rezultat testów i nauka dla nas

PARASOFT

- Klient w dalszym ciągu używa naszego rozwiązania.
  - U klienta udało się osiągnąć blisko 18-krotne przyśpieszenie.
- Mamy środowisko, które używamy do testów.
- Koszt diagnozy vs. utrata kontraktu

Wnioski

- Automatyzacja testów nie jest za darmo, dlatego często jest pomijana. Jednak zdecydowanie oplaca się ją robić.
- Nie wszystkie testy dobrze się automatyzują, np. z UI są zwykle kłopoty, natomiast API automatyzuje się świetnie.
- Szczególnie testy niefunkcjonalne powinny być automatyzowane, bo to na nie zwykle brakuje czasu gdy terminy gonią.

Pytania ?

PARASOFT

???