

Artur Jankowski indeks: 317928

1)

Miary:

Ile zapytań system potrafi przetworzyć na jednostkę czasu?

Pomiar: $X = A / T$

Dane:

T -> czas działania

A -> liczba przetworzonych zapytań

Ograniczenia: $0 \leq X \leq 1$

Cel: Im bliżej 1.0, tym lepiej

Przepustowość pobierania/wysyłania

Pomiar: $X = A / T$

Dane:

T -> czas pobierania/wysyłania

A -> rozmiar pliku

Ograniczenia: $0 \leq X$

Cel: Im mniej, tym lepiej

Ile użytkowników może korzystać jednocześnie z systemu?

Pomiar: $X = A / T$

Dane:

T -> czas działania

A -> liczba jednoczesnych dostępów

Trwałość bazy danych

Pomiar:

$X = 1 - E / N$

Dane:

E -> liczba uszkodzeń danych

N -> łączna ilość testów

Ograniczenia: $0 \leq X \leq 1$

Cel: Im bliżej 1.0 tym lepiej

Przypadek testowy

ID: UC-123

Tytuł: Przetworzenie zapytania o dużym rozmiarze

Aktor: Użytkownik

Warunek początkowy: Użytkownik jest upoważniony do przesyłania danych

Warunek końcowy: Dane zostały przesłane na serwer

Główny scenariusz:

1. Użytkownik przesyła na serwer plik load_test.zip o rozmiarze 100 GB z załączoną osobno sumą kontrolną.
2. System odbiera plik w częściach. Osobno otrzymuje sumę kontrolną.
3. Na serwerze znajduje się plik load_test.zip.
4. Plik load_test.zip jest poprawnie weryfikowany przez sumę kontrolną.

2)

Techniki:

Sprint – interwały trwające maksymalnie jeden miesiąc, w których trakcie, wytwarza się kolejny Increment. W trakcie nich odbywa się: planowanie Sprintu, codzienne Standupy, ocena Sprintu itp. W założeniu wszystkie działania powinny dążyć do osiągnięcia celu Sprintu według Definition of Done. Na koniec Sprintu, zespół powinien pokazać interesariuszom coś gotowego, jakiś widoczny efekt,

Daily Scrum/Standup – w założeniu krótkie spotkanie, odbywające się codzienne o tej samej porze, na którym członkowie zespołu informują o postępach w swoich zadaniach, o tym co ukończyli, albo czym planują się zająć,

Sprint Review – zespół prezentuje klientowi postępy prac oraz efekty Sprintu, jest na nim często modyfikowany Product Backlog,

Sprint Retrospective – ma na celu wykazanie błędów albo elementów o niskiej efektywności dotyczących komunikacji, organizacji pracy, interakcji, narzędzi. Określa się na nim, co można wprowadzić, by zwiększyć efektywność.

Narzędzia:

Product Backlog – zmieniająca się, uporządkowana lista tego, co jest konieczne do ulepszenia produktu,

Sprint Backlog – cel Sprintu oraz plan dostarczenia,

Tablica zadań – wizualizowanie obecnych zadań i ich stanu,

Wykres np. wypalenia – prognozowanie postępów.

Weryfikacja:

Proces weryfikacji i walidacji jest oparty na Definicji Ukończenia dla poszczególnych zadań w Backlogu oraz na pokazywaniu wartości dodanej w kolejnych Incrementach interesariuszom na Sprint Review. Proces ten ułatwia podział celu na mniejsze części procesu doskonalenia zwane Incrementami. Ponadto stosuje się Unit Testy, Testy integracyjne, Code Review i inne praktyki zarządzania jakością.

3)

Lead by:

Example – stosujemy się do tego, czego wymagamy od innych np. jeśli wymagamy punktualności, sami się nie spóźniamy, nie zlecamy komuś zadania, którego sami nie chcielibyśmy wykonać, jeśli wymagamy stosowania się do przepisów firmy, sami też się do nich stosujemy itp.

Transparency – ważna jest przejrzystość oraz łatwa możliwość komunikacji w zespole. Zadania mają przydzielone osoby, które są za nie odpowiedzialne, zmiany w kodzie mają autora itp. Przejrzystość ułatwia inspekcję, a co za tym idzie, ogranicza błędy i straty

Empathy – w myśl Toyody, korzystanie z opinii ludzi na miejscu w ulepszaniu procesów, np. możemy zatrudnić użytkownika, który proponuje korzystne dla nas zmiany, aby towarzyszył w ich wdrożeniu.

4)

Członek zespołu wytwarzającego oprogramowanie wg. metodyki Scrum:

- wybiera zadania do wykonania w obecnym sprincie z Product Backlogu do Sprint Backlogu, w zależności od swoich umiejętności (np. dla wdrażanego Juniora zadania może przydzielić bardziej doświadczony programista), ich priorytetu oraz przewidywanej pracochłonności,
- wykonuje zadania na podstawie ich definicji ukończenia,
- dąży do osiągnięcia celu Sprintu, adaptuje się do różnych ról,
- jest odpowiedzialny za swoje zadania, wymusza odpowiedzialność zawodową na współpracownikach za ich zadania,
- komunikuje się z resztą zespołu deweloperskiego oraz Scrum Masterem,
- wyróżnia go szeroki zakres umiejętności oraz brak specjalizacji,
- samoorganizuje Sprint wraz z resztą zespołu.

5)

Product Owner – jest odpowiedzialny za maksymalizację wartości produktu wytwarzanego przez Scrum Team

Obowiązki:

- tworzy product backlog, uporządkowuje go, nadaje priorytety, sprawia, by był przejrzysty, czytelny,
- komunikuje się z biznesem, interesariuszami klienta itp. by tworzyć wizję i mapę rozwoju,
- może wydłużyć/skrócić sprint, lub go przerwać,
- odpowiedzialny za realizację wizji produktu.

Narzędzia:

- Product Backlog, priorytetyzowanie odpowiednich zadań,
- historyjki użytkownika,
- narzędzia raportowania, analizy itp.,
- prototypy, demo technologiczne,
- tablica koncepcyjna, wizja produktu,
- ścieżka/mapa rozwoju.

Scrum Master - pilnuje stosowania zasad metodyki Scrum oraz efektywności zespołu

Obowiązki:

- stara się, by wydarzenia związane ze Scrumem były produktywnie oraz nie przekraczały wyznaczonych wcześniej ram czasowych,
- komunikuje zmiany w Backlogu,
- szkoli w zakresie wdrażania Scruma,
- raportuje o wyniku Sprintu,
- śledzi postępy zespołu w Sprincie oraz usuwa bariery je ograniczające oraz te pomiędzy klientem a zespołem.

Narzędzia:

- oprogramowanie np. Jira, Slack,
- Sprint Backlog,
- tablica zadań,
- różnego rodzaju wykresy tj. burn-down, burn-up, cumulative flow, velocity.