

## Dokument roboczy projekt ZSI Moduł 2

Arkadiusz Kałuża, Adam Kierat

Przygotowanie dokumentu roboczego zawierającego zapisy dotyczące punktów wyróżnionych we wzorze dokumentu roboczego dostępnego na platformie zdalnej edukacji.

- **A1. Ustanowienie współdzielonego repozytorium**

Ustanowienie współdzielonego repozytorium na platformie github.com      Link:  
<https://github.com/arcziko9/ZarzadzanieSystemamiInformatycznymiPOLSL>

- **B1. Ustalenie z prowadzącym tematu projektu.**

Sekcja wybrała projekt numer 2.4:

- a. Scharakteryzuj istniejący (dowolnie wybrany) system informatyczny wspomagający pracę firmy programistycznej
- b. Przedstaw (na przykładzie) praktyczne aspekty analizy dostępności serwisów internetowych (*high-availability*)
- c. Opracuj wizualizację (w postaci → mapy myśli) standardu dokumentacji systemu informatycznego

- **B3. Przygotowanie notatki z ASD: 3 pozycje, 3 najważniejsze wyróżniki, linki do witryn zawierających szczegóły.**

a. GitLab to menadżer repozytoriów oparty o architekturę Gita, a także potężne narzędzie służące do rozwoju oprogramowania. Wykorzystując przyjazny interfejs użytkownika, GitLab umożliwia wydajną pracę, zarówno przy użyciu wiersza poleceń, jak i samego UI. GitLab jest bardzo pomocny zarówno dla deweloperów, jak i również dla innych członków zespołu, zapewniając wszystkim pracownikom pojeдинczą, unikatową platformę roboczą.

- \* <https://ujeb.se/GGq2h2>

- \* <https://ujeb.se/Cz7Ha>

- \* <https://ujeb.se/RTcyB>

- b. **System wysokiej dostępności** – systemy informatyczne charakteryzujące się odpowiednio dostosowywaną: niezawodnością, dostępnością, wydajnością do specyficznych, zwykle krytycznych, zastosowań danego systemu. System komputerowy klasyfikowany jest jako wysokiej dostępności, jeśli jest niedostępny przez czas rzędu 5 minut w roku (dostępność od 99,999% , a mniej niż 99,9999% czasu).

- \* <https://ujeb.se/QITna>

- \* <https://ujeb.se/8k1qA>

- \* <https://ujeb.se/qqXDX>

- **C1.Opracowanie koncepcji realizacji projektu.**

**Workflow GitLab’a** to nic innego jak sekwencja działań podczas całego procesu tworzenia oprogramowania z wykorzystaniem platformy GitLab.

Workflow GitLab’a oparty jest o GitLab Flow i wspiera tym samym zarządzanie wersjonowaniem zgodnie z zasadami Git’a. Jego głównym celem jest zintegrowanie pracy zespołów i poprawa wydajności na wszystkich etapach – od implementacji do produkcji.

### **Etapy rozwoju oprogramowania**

○ \*

Proces rozwoju oprogramowania w tradycyjny sposób przebiega najczęściej w 10-ciu głównych krokach. GitLab stworzył rozwiązanie dla każdego z nich:

- \* **POMYSŁ:** Każda nowa propozycja zaczyna się od pomysłu, z reguły wyrażanego w rozmowie. Na tym etapie, GitLab wykorzystuje integrację z Mattermost.
- \* **ZGŁOSZENIE:** Najbardziej wydajnym sposobem na omówienie pomysłu jest zgłoszenie go w systemie. Twój zespół i współpracownicy mogą pomóc Ci w jego dopracowaniu i rozwoju przy pomocy Issue Trackera.
- \* **PLAN:** Po zakończeniu procesu konsultacji, zaczyna się czas kodowania. Najpierw jednak trzeba nadać priorytety i zorganizować workflow. W tym celu możemy użyć Issue Board.
- \* **KODOWANIE:** Gdy zakończymy planowanie, możemy zacząć pisać kod.
- \* **COMMIT:** Gdy wersja robocza jest już satysfakcjonująca, możemy commitować kod do funkcjonalnego brancha z kontrolą wersji.
- \* **TESTY:** Dzięki GitLab CI możemy uruchamiać skrypty budujące i testujące naszą aplikację.
- \* **RECENZJA:** Gdy nasze skrypty, buildy i testy działają, jesteśmy gotowi do recenzji kodu i akceptacji.
- \* **STAGING:** Teraz czas na wysłanie kodu do środowiska testowego, aby sprawdzić czy wszystko działa poprawnie.
- \* **PRODUKCJA:** Gdy wszystko działa prawidłowo, czas na wdrożenie w środowisku produkcyjnym.
- \* **FEEDBACK:** Ostatni etap to miejsce na wprowadzanie poprawek. Możemy wówczas również wykorzystać narzędzie Cycle Analytics, aby sprawdzić ile czasu poświęciliśmy na każdy z etapów.
- \* Efektywne przejście przez wszystkie wymienione etapy wymaga dobrych mechanizmów wspierających workflow. W kolejnych sekcjach zamieściliśmy opis zestawu narzędzi ułatwiających pracę i przygotowanych przez GitLab.



IDEA



ISSUE



PLAN



CODE



COMMIT



TEST



REVIEW



STAGING



PRODUCTION



FEEDBACK

- Dokumentacja jest potrzebna. Jest częścią produktu, ma wartość biznesową - pozwala ograniczyć koszty i pozyskiwać nowych klientów. W zależności od zastosowania, odbiorcy czy skali projektu, może ona przybierać różne formy. Zazwyczaj pisanie dokumentacji nie wymaga tworzenia tysięcy dokumentów i skomplikowanego języka - treści powinny być proste i zrozumiałe dla jak największej grupy odbiorców. W idealnej sytuacji, dokumentacja może być częścią procesu tworzenia oprogramowania, co pozwala ograniczyć czas potrzebny na jej przygotowanie i publikację.

powinniśmy zasięgnąć trochę wiedzy na temat wypracowanych metod skutecznej komunikacji. Dobrym przykładem będzie **plain language** - czyli zestaw wytycznych, które mówią wprost, co to znaczy, że tekst jest prosty. Celem jest tworzenie dokumentów w taki sposób, aby były zrozumiałe dla możliwie szerokiego spektrum odbiorców.

\* **Przykładowe narzędzia do tworzenia dokumentacji technicznej:**

† **Natural Docs** ( <https://www.natualdocs.org/> )

pozwała dokumentować kod napisany w jednym z 21 języków programowania, a ponadto można go łatwo rozszerzyć o więcej, więc bez względu na to, czego używasz, może również. A jeśli twój projekt używa wielu języków, nie ma problemu! Wszystko to będzie zawarte w tym samym zestawie dokumentacji.

† **DoxyGen** ( <http://www.doxygen.nl/> )

Ndo generowania dokumentacji ze źródeł C ++ z adnotacjami, ale obsługuje także inne popularne języki programowania, takie jak C, Objective-C, C # , PHP, Java, Python, IDL (smaki Corba, Microsoft i UNO / OpenOffice) ),

- Opisywaną alternatywą do Asana będzie aplikacja o nazwie Trello.

Trello umożliwia wizualne zarządzanie notatkami, czy raczej jak mówią twórcy — „wszystkim” i dzielenie się tym ze „wszystkimi” . Nieważne czy pracujesz solo, czy z zespołem, Trello ma tę cudowną zaletę, że jest typowym narzędziem. To, co z nim zrobisz zależy tylko od Ciebie oraz Twoich współpracowników.

Przykładowe zalety trello:

- Codzienna organizacja - jeśli Twój dzień jest wypełniony różnego rodzaju zadaniami, posiadanie wszystkich tych zadań w jednym miejscu może być niezbędne. Trello może zestawiać wszystkie twoje zadania w listy i nadal pozwala ci uzyskać doskonały widok na cały dzień.
- Śledzenie czasu - Śledzenie czasu niekoniecznie musi być tym, które ładuje się z godziny. To naprawdę wspaniały sposób na zwiększenie wydajności.
- Czytelna lista zadań - Jeśli masz powoli rosnący katalog książek, które chcesz przeczytać, możesz po prostu przechowywać listę w Trello, tworząc karty dla każdej z nich.

- Łatwe planowanie wakacji - Planowanie wakacji zazwyczaj wymaga wielu decyzji. Trello pomaga to uprościć, tworząc listy codziennych czynności.
- Zarządzaj budżetem - Trello może pomóc w stworzeniu planu finansowego. Możesz skonfigurować karty i śledzić dochody i wydatki, kiedy ich potrzebujesz.
- Zarządzanie projektami - niezależnie od projektu, Trello będzie do Twojej dyspozycji. Utworzenie tablicy z indywidualnymi listami może zarządzać wszystkimi małymi zadaniami, które są częścią twojego ogólnego projektu. Trello pomaga to uprościć, łączy listy codziennych operacji.

#### Wady aplikacji Trello:

- Nie ma wsparcia offline - głównym problemem nie tylko Trello, ale ogólnie tego rodzaju oprogramowania jest poleganie na danych. Jeśli nie masz dostępu do Internetu, nie masz dostępu do swojego Trello. Dostęp do danych może nie wydawać się problemem, biorąc pod uwagę wiek, w którym żyjemy, ale niestety zawsze będą sytuacje, w których nie będzie można uzyskać dostępu do danych.
- Pamięć jest ograniczona - chociaż Trello może przechowywać wiele załączników, załączniki te są ograniczone do 250 MB na przesyłanie, jeśli jesteś złotym członkiem. Problem polega na tym, że masz tylko 10 MB na przesyłkę, jeśli jesteś podstawowym członkiem.
- Komentowanie - ciągłym problemem związanym z Trello, który należy naprawić, jest to, że nie można edytować komentarza na karcie. Po opublikowaniu i zapisaniu komentarza pojawi się opcja napisania nowego komentarza zamiast edycji oryginalnego.
- Zarządzanie dużymi projektami bywa problematyczne - jeśli chodzi o mniejsze projekty, Trello ma swoje. Jeśli jednak zarządzanie projektami na większą skalę jest wymagane, być może Trello nie jest najlepszą opcją.

#### Systemy wysokiej dostępności przykłady:

- Oprogramowanie HA – bazy danych

Dane przechowywane w bazach stanowią zwykle podstawę działania wielu aplikacji. Bez dostępu do nich usługi są bezużyteczne. Dlatego też producenci najpopularniejszych silników baz danych wyposażyli swoje produkty w mechanizmy zapewniające wysoką dostępność i minimalizujące skutki potencjalnych awarii.

Dla Microsoft SQL zaprojektowano kilka rozwiązań, które zapewniają wysoką dostępność zarówno poszczególnych baz danych, jak i całych instancji, np. klastrowanie SQL. Przy takiej konfiguracji ochronie podlegają wszystkie elementy serwera SQL, a nie tylko poszczególne bazy. Klastrowanie SQL opiera się w głównej mierze na funkcji dostępnej w samym systemie Windows Server, dlatego należy je skonfigurować na tym poziomie. SQL Clustering możliwy jest zarówno w wersji Standard,

jak i Enterprise SQL serwera, z tą różnicą, że pierwsza z nich umożliwia jedynie konfigurację active-passive.

**Konfiguracja Log shipping** jest rozwiązaniem, które wykorzystuje funkcjonalność dziennika transakcji. Polega na cyklicznym wykonywaniu kopii dziennika i ich odtwarzaniu na zapasowym serwerze.

**Mirroring**, czyli mechanizm lustrzanej kopii baz, jest funkcją dość prostą w implementacji. Polega na synchronicznym transferze i odtwarzaniu transakcji między bazą podstawową i zapasową, co zapewnia, że dane po obu stronach zawsze są aktualne.

Do elementów konfiguracji wysokiej dostępności Microsoft SQL Server należy również rozwiązanie **AlwaysOn**. Polega ono na utrzymywaniu grupy instancji, na których znajdują się repliki bazy. Takich replik może być maksymalnie osiem, z czego pierwsza (podstawowa) stanowi bazę produkcyjną, a do pozostałych (zapasowych) trafia kopia danych w postaci logu transakcyjnego.

- **Oprogramowanie HA – systemy poczty elektronicznej**

Wykorzystanie mechanizmów wysokiej dostępności w systemach obsługi poczty elektronicznej bardzo dobrze widać na przykładzie Microsoft Exchange Servera, gdzie użyte funkcje skutecznie pozwalają na utrzymanie ciągłości świadczonych usług.

O wysoką dostępność serwerów Mailbox, które przechowują bazy danych z pocztą użytkowników, dba funkcja Database Availability Groups. Stanowi ona grupę serwerów opartych na technologii klastra pracy awaryjnej. Ich konfigurację i dodawanie kolejnych nodów w prosty sposób wykonuje się z konsoli zarządzania Microsoft Exchange. Warto również nadmienić, że do poprawnego działania DAG nie wykorzystuje się wspólnej przestrzeni dyskowej. Każdy z serwerów objętych konfiguracją DAG przechowuje kopie bazy, która jest aktualizowana dzięki ciągłej replikacji logów transakcyjnych pomiędzy wszystkimi serwerami Data Availability Group. Usługa Microsoft Exchange Replication czeka, aż log transakcyjny aktywnej bazy zostanie zapisany i zamknięty. Dopiero wtedy następuje replikacja do wszystkich pasywnych kopii baz. Po sprawdzeniu poprawności logów usługa Information Store odtwarza je i wtedy następuje ostateczna synchronizacja pasywnych kopii baz.

- **Sprzęt HA – równoważenie obciążenia**

Oprócz funkcji wysokiej dostępności w systemach operacyjnych lub aplikacjach na rynku dostępne są również urządzenia zapewniające utrzymanie wysokiej dostępności usług w infrastrukturze IT. Konfiguracje High Availability, analogicznie jak w przypadku systemów, występują też w postaci sprzętowych klastrów pracy awaryjnej lub urządzeń równoważących obciążenie.

Jednym z producentów sprzętowych rozwiązań jest firma Barracuda Networks z urządzeniem Barracuda Load Balancer ADC. Pozwala ono równoważyć obciążenie

serwerów sprzętowych oraz maszyn wirtualnych obsługujących dowolną liczbę protokołów warstwy 4 oraz warstwy 7 (m.in. HTTP/S, SMTP, IMAP, FTP, DNS, TCP, UDP). Realizacja load balancingu może opierać się na sieciach IPv4, IPv6 lub mieszanych, gdzie istnieje konieczność translacji adresów.

- **Nieprzerwany dostęp do internetu**

W wielu przypadkach ważnym elementem funkcjonowania firmy jest nieprzerwany dostęp do internetu. Obecnie wielu dostawców urządzeń dostępowych, routerów i urządzeń UTM umożliwia korzystanie z kilku łączy do różnych dostawców. Może to być dostęp realizowany po sieci Ethernet, jak również z wykorzystaniem modemów GSM. Takie urządzenia ma w ofercie Cisco (np. seria urządzeń SonicWall TZ) i Fortinet (urządzenia Fortigate).

Konfiguracja dwóch łączy pozwala na utrzymanie redundancji połączenia internetowego. W razie awarii łącza od jednego dostawcy ruch kierowany jest na drugą bramę internetową. Zwykle przy dostępnych wielu łączach wspierane są dwie konfiguracje. Pierwsza to utrzymywanie drugiego łącza jako zapasowego, w którym ruch kierowany jest jedynie w przypadku awarii łącza podstawowego. Natomiast druga konfiguracja pozwala na zbalansowanie ruchu pomiędzy dwoma łączami. Wówczas urządzenie potrafi w odpowiedni sposób sterować połączeniem, tak aby zestawione sesje były kierowane zawsze właściwym łączem. Konfiguracja dwóch aktywnych łączy wykorzystywana jest również w przypadku, gdy istnieje potrzeba rozdzielenia konkretnego rodzaju ruchu (np. HTTP i SMTP) pomiędzy dwóch dostawców, żeby nie obciążać pojedynczego łącza. Wówczas jeśli nastąpi awaria, cały ruch zostanie przekierowany do sprawnego połączenia.

- **Niezawodność pamięci masowych**

Macierze dyskowe i serwery NAS umożliwiają zachowanie wysokiej dostępności dla funkcji przetwarzania danych na wielu płaszczyznach. W ich przypadku występuje zasada, że im „wyższy”, bardziej zaawansowany model, tym więcej rozwiązań zapewniających bezpieczeństwo poprawnej pracy. Podstawą ochrony i dostępu do plików jest utworzenie odpowiedniego poziomu RAID (co najmniej 1), jednak dla uzyskania większego bezpieczeństwa można skonfigurować choćby RAID 6, który pozwoli na awarię do dwóch dysków bez utraty danych.

Dodatkiem do stałego dostępu do pamięci masowej są nadmiarowe porty sieciowe. W przypadku serwerów NAS już w modelach z dolnej półki można spotkać sprzęt wyposażony w dwa interfejsy sieciowe, co w połączeniu z co najmniej dwoma przełącznikami pozwoli na uzyskanie niezależności od awarii jednego z nich.

- **Wirtualizacja**

W przypadku rozwiązań wirtualizacyjnych administratorzy mają do dyspozycji szereg funkcji umożliwiających zachowanie najwyższego poziomu dostępności usług. Co prawda, zbudowanie bezpiecznego środowiska chmury wiąże się również z rozwiązaniami, o których pisaliśmy wcześniej (klastrowanie, HA pamięci masowych), to oprogramowanie do wirtualizacji pozwala na integrację tych rozwiązań z dodatkiem

własnych funkcji. Bez względu, czy jest to Hyper-V Microsoftu, VMware VSphere czy inny wirtualizator, wszędzie możemy znaleźć szeroki wachlarz funkcji tworzących bezpieczne, wysoko dostępne środowisko. Różnić je może jedynie cena zakupu oraz sposób implementacji w danym środowisku.

- Mapa myśli standardu dokumentacji systemu informatycznego

