|  |  |  |  |
| --- | --- | --- | --- |
| **Karta projektu badawczo-rozwojowego** | | | |
|
|
|
| **Tytuł projektu** | | | |
| **Opracowanie i wdrożenie systemu informatycznego wsparcia IT/OT do obsługi i priorytetyzacji zgłoszeń** | | | |
| **Numer ewidencyjny projektu** | | **BR – zarządzanie zadaniami** | |
| **OPIS DZIAŁAŃ BADAWCZO ROZWOJOWYCH:** | | | |
|
| ***Cel/ Opis nowych zakładanych właściwości/ funkcjonalności rozwiązania (produktu lub procesu)*** | Celem projektu: zaprojektowanie i wdrożenie cyfrowego systemu wsparcia IT/OT pozwalającego na:   * Priorytetyzację zadań – zgłoszeń do obsługi, * zminimalizowanie czasu reakcji zespołu wsparcia IT na awarie krytyczne (postój linii produkcyjnych, incydenty produkcyjne), * porządkuje komunikację między produkcją a działem IT oraz * zapewnia pełną śladowość i raportowanie.   Rozwiązanie oparto na GLPI (Gestionnaire Libre de Parc Informatique) – otwarto-źródłowym oprogramowaniu klasy ITSM/Service Desk do obsługi zgłoszeń, zarządzania zasobami (CMDB), umowami SLA i bazą wiedzy – przy czym konfigurację, rozszerzenia funkcjonalne i integracje opracowano we własnym zakresie. Dodatkowo do oprogramowania zaimplementowano narzędzie Pushover - usługa wysyłania powiadomień „push” na telefony (iOS/Android) i komputery. Daje prosty interfejs REST oraz aplikację mobilną/desktop, dzięki czemu systemy (np. GLPI, ERP, skrypty) mogą natychmiast wysyłać alerty do wybranych osób lub grup.  Opracowane właściwości i funkcjonalności dotyczyły:   * Wielokanałowy intake zgłoszeń: e-mail, terminale stanowiskowe/CIM na hali, formularze w ERP; automatyczne dołączanie kontekstu produkcyjnego (linia/stanowisko/zlecenie). * Automatyczna klasyfikacja i priorytetyzacja (P1–P4) z regułami routingu do właścicieli usług; deduplikacja i łączenie incydentów. * Powiadomienia natychmiastowe dla P1/P2 (push przez Pushover) i kaskadowe eskalacje SLA; harmonogram dyżurów (on-call). * CMDB kluczowych zasobów IT/OT z historią incydentów, zmian i przestojów; powiązanie zgłoszeń z aktywami i umowami serwisowymi. * Baza wiedzy i runbooki (szablony działań) dla przypadków powtarzalnych; automatyczne podpowiedzi w formularzach zgłoszeń. * Raportowanie KPI: MTTA, MTTR, FCR, zgodność z SLA; pulpity dla produkcji i IT; eksport do analiz kontrolingowych. * Integracje opracowane wewnętrznie (REST/API, webhooki): dwukierunkowa wymiana statusów z ERP oraz CIM; automatyczne zakładanie/aktualizacja zgłoszeń z alarmów procesowych. * Bezpieczeństwo i audyt: role/uprawnienia, rejestr zmian, polityki retencji danych; wdrożenie on-premises.   System oprogramowano, skonfigurowano i rozszerzono we własnym zakresie (wtyczki, workflow, szablony, integracje), tak aby odpowiadał specyfice produkcji seryjnej i pozwalał na natychmiastowe wyłuskanie oraz obsługę zgłoszeń krytycznych, udrożniając komunikację w razie zatrzymania procesu produkcyjnego.  Projekt był prowadzony metodyką iteracyjną w sprintach (1–2 tygodnie) z backlogiem wymagań i kryteriami akceptacji: MVP, następnie integracja CIM, integracja ERP, uruchomienie bazy wiedzy, wdrożenie SLA/raportów. Przyrost był demonstrowany użytkownikom, testowany na wybranych liniach (UAT), a wnioski z retrospektyw zasilały kolejne iteracje. Definicja ukończenia obejmowała: działające reguły eskalacji, scenariusze testów P1, komplet runbooków i metryki MTTA/MTTR dostępne na kokpicie. | | |
|
|
|
| ***Podstawowe etapy projektu*** | | | |
| ***Numer etapu*** | ***Nazwa etapu*** | | | ***Data realizacji*** |
| 1. | Zdefiniowanie wymagań krytyczny i Inicjalizacja | | | 09.2024 |
| 2. | Opracowanie wersji MVP Serivce Desku | | | 10.2024 |
| 3. | Automatyzacja i krytyczne alerty | | | 10.2024 –  11.2024 |
| 4. | Skalowanie i ciągłe doskonalenie | | | 12.2024 |
| ***Wykaz najważniejszych problemów badawczych oraz sposób ich rozwiązania*** | 1. Problem: nadmiar i duplikacja zgłoszeń (niski signal-to-noise), brak możliwości szybkiego „wyłuskania” spraw krytycznych. Rozwiązanie: opracowano reguły korelacji i deduplikacji (matching nadawcy, zasobu CMDB, sygnatury błędu), wprowadzono semantyczne tagowanie i ontologię usług/komponentów, a także automatyczne łączenie wątpliwych duplikatów w GLPI. 2. Problem: brak wiarygodnej CMDB i widoczności zależności usługowych (co utrudniało właściwą eskalację i ocenę wpływu). Rozwiązanie: utworzono CMDB w GLPI (klasy CI: serwer, stacja, skaner, linia/gniazdo, aplikacja, interfejs), odwzorowano relacje usługa-komponent, powiązano CI z ticketami i SLA; uruchomiono procedury inwentaryzacyjne i cykliczne przeglądy danych. 3. Problem: opóźnienia i zawodność powiadomień o awariach produkcyjnych. Rozwiązanie: wdrożono Pushover jako kanał krytyczny z redundancją e-mail/SMS, listami dyżurowymi i oknami serwisowymi; dodano testy heartbeat oraz monitorowanie dostarczalności (delivery/ACK) i reguły ponowień. 4. Problem: brak bazy wiedzy i KEDB (Known Error Database) – wysoka zależność od wiedzy niejawnej operatorów. Rozwiązanie: zaadaptowano praktykę KCS w GLPI: szablony artykułów, runbooki/playbooki dla zdarzeń P1/P2, wersjonowanie treści, powiązania KB↔ticket; wdrożono progi jakości (recenzje, wskaźnik ponownego użycia KB). 5. Problem: brak mierzalności procesu (MTTA, MTTR, FCR, reopen rate) i trudność w wykrywaniu wąskich gardeł. Rozwiązanie: zdefiniowano KPI i cykle raportowe; zbudowano pulpity operacyjne GLPI z metrykami MTTA/MTTR per kategoria/linia/zmiana, raporty SLA, trend powtarzalnych incydentów; automatyczne raporty dzienne do właścicieli usług. 6. Problem: bezpieczeństwo i zgodność – niejednoznaczne uprawnienia, brak ścieżki audytu i polityk retencji. Rozwiązanie: wprowadzono RBAC w GLPI, SSO dla użytkowników wewnętrznych, pełne logi audytowe, polityki retencji i maskowania danych wrażliwych; segregacja obowiązków między 1./2./3. linią wsparcia. 7. Problem: ryzyko zatrzymania procesu w razie niedostępności narzędzia lub strefy bez łączności. Rozwiązanie: procedury pracy w trybie degradacji (formularze offline, kolejkowanie lokalne, rejestr awaryjny), mechanizmy buforowania i retransmisji zgłoszeń po powrocie łączności; testy DR dla GLPI. 8. Problem: zarządzanie zmianą konfiguracyjną narzędzia i integracji (ryzyko regresji przy modyfikacjach reguł/plug-inów). Rozwiązanie: uruchomiono proces RFC/Change w GLPI z CAB, środowiska DEV/TEST/PROD i lekki CI/CD konfiguracyjny (wersjonowanie reguł, szablonów, słowników), testy UAT z kryteriami DoR/DoD. 9. Problem: ergonomia zgłoszeń po stronie użytkowników produkcji – nadmierna liczba pól i błędy klasyfikacji. Rozwiązanie: zaprojektowano uproszczone formularze per rola/stanowisko, predefiniowane kategorie usług, walidacje kontekstowe, podpowiedzi i autouzupełnianie CI; szkolenia mikro + feedback loop z retrospektyw. 10. Problem: brak szybkiej identyfikacji spraw powtarzalnych i przyczyn źródłowych. Rozwiązanie: mechanizmy grupowania wg wzorca (signature-based), tagowanie problemów, proces Problem Management z KEDB i działaniami korygującymi; przeglądy Pareto i backlog problemów technicznych w sprintach. | | |
| ***Podstawowe prace o charakterze twórczym w projekcie*** | 1. Zaprojektowanie architektury usługi Service Desk opartej na GLPI (ITSM/CMDB/KB/SLA) – model procesu przyjęcia–diagnozy–eskalacji–zamykania w notacji BPMN 2.0, wraz z przepływami dla incydentów P1/P2 i zmian RFC/Standard Change. 2. Opracowanie metamodelu CMDB dla środowiska produkcyjnego: klasy CI (linia/gniazdo, maszyna, sterownik PLC/HMI, aplikacje ERP/MES, stacje robocze, interfejsy sieciowe), atrybuty, relacje zależności usługowych oraz powiązanie CI↔SLA↔ticket. 3. Zdefiniowanie taksonomii zgłoszeń i macierzy priorytetów (impact × urgency) oraz katalogu SLA/SLO; implementacja w GLPI reguł priorytetyzacji, kalendarzy serwisowych i czasów reakcji/naprawy dla klas P1–P4. 4. Stworzenie warstwy integracyjnej kanałów wejściowych: • konektor IMAP/SMTP do kolejkowania e-maili, • webhook/REST z ERP do automatycznego tworzenia zgłoszeń, • lekki klient HMI dla stanowisk CIM. Opracowano schemat normalizacji pól (requester, asset, kategoria, impact/urgency) i idempotentne reguły tworzenia/aktualizacji ticketów. 5. Implementacja algorytmów korelacji i deduplikacji zgłoszeń (sygnatury błędów, matching CI, porównanie treści/tematu, okna czasowe), w tym automatyczne łączenie duplikatów oraz reguły „problem clustering” do inicjowania zadań Problem Management. 6. Zaprojektowanie i wdrożenie kanału powiadomień krytycznych w oparciu o Pushover: profile alarmowe (emergency), parametry retry/expire, potwierdzenia ACK, listy dyżurowe L2/L3, reguły eskalacji kaskadowej i monitorowanie dostarczalności (delivery/ACK). 7. Utworzenie bazy wiedzy (KB) i KEDB zgodnie z praktyką KCS: szablony runbooków/playbooków dla typów incydentów, powiązania KB↔ticket/CI, wersjonowanie treści i miary ponownego użycia artykułów. 8. Budowa pulpitów operacyjnych i raportowania: MTTA/MTTR, FCR, reopen rate, wykonanie SLA, Pareto powtarzalnych usterek, obciążenie kolejek; automatyczne raporty cykliczne (dzienny/tygodniowy) dla właścicieli usług. 9. Projekt bezpieczeństwa i zgodności: RBAC w GLPI, integracja z SSO, segmentacja ról (Użytkownik/Service Desk/L2/L3/Change Manager), rejestry audytowe, polityki retencji danych oraz maskowanie informacji wrażliwych. 10. Przygotowanie trybu pracy w degradacji (DR/BCP): formularze offline dla P1, buforowanie zgłoszeń przy braku łączności, procedury retransmisji, kopie konfiguracji i testy odtworzeniowe narzędzia. 11. Ux/UI i ergonomia formularzy: kontekstowe formularze zgłoszeń per rola/stanowisko, walidacje wymaganych pól, słowniki podpowiedzi, automatyczne przypisanie CI po skanie/wyborze, minimalizacja kroków użytkownika. 12. Pipeline zmian konfiguracyjnych (CI/CD dla GLPI): środowiska DEV/TEST/PROD, repozytorium reguł i szablonów (versioning), proces RFC z kryteriami DoR/DoD, testy UAT dla nowych integracji i wtyczek. 13. Pakiet szkoleniowy i adopcyjny: materiały dla użytkowników i zespołów L1–L3, checklisty operacyjne, scenariusze ćwiczeń P1 (war-game), cykl retrospektyw sprintowych i backlog usprawnień. 14. Kompletny korpus dokumentacji technicznej: katalog usług i SLA, model CMDB, specyfikacje interfejsów (API/webhook), polityki eskalacji, runbooki, procedury DR oraz instrukcje administracyjne GLPI. | | |
| ***Poziom innowacyjności projektu*** | **Innowacja w skali przedsiębiorstwa** | | **Innowacja w skali kraju** |
| Tak | | Nie |
| ***Podsumowanie projektu*** | W ramach prac rozwojowych opracowano zintegrowany, parametryzowalny system obsługi zgłoszeń – oparty na platformie GLPI, rozszerzony o własne reguły korelacji, integracje z ERP/CIM i krytyczne powiadomienia Pushover – który w sposób mierzalny skrócił MTTA/MTTR dla incydentów produkcyjnych, podniósł przejrzystość kolejek, umożliwił systematyczne usuwanie przyczyn źródłowych i zapewnił audytowalność oraz bezpieczeństwo procesu wsparcia. | | |
|
|
|
|
|
|
|
|
|
| **Dokumentacja projektowa (załączniki do karty projektu)** | | | |
| 1. | Kod źródłowy programu - repozytorium na serwerze | | |
| 2. | Materiały dla użytkowników i zespołów | | |
| 3. | Dokumentacja techniczna: katalog usług i SLA, model CMDB, specyfikacje interfejsów (API/webhook), polityki eskalacji, runbooki, procedury DR oraz instrukcje administracyjne GLPI | | |