



LEAN ARCHITECTURE FRAMEWORK

for Continuous Transformation

Version 1.2

Autorzy

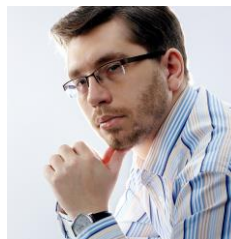


Tomasz Michałek-Czerepak



Tomasz Świerszcz

Współpraca



Andrzej Sobczak

©2020 LAF Institute. Dokument publikowany jest na licencji Creative Commons Uznanie autorstwa-Na tych samych warunkach. Uproszczona treść licencji w języku polskim: <https://creativecommons.org/licenses/by-sa/4.0/deed.pl>, pełne brzmienie licencji w języku angielskim: <https://creativecommons.org/licenses/by-sa/4.0/legalcode>. Korzystając z LAF, potwierdzasz fakt zapoznania się i wolę przestrzegania treści tej licencji.



Słownik

LAF korzysta z trzech podstawowych koncepcji, które w zależności od kontekstu w świecie IT mają wiele znaczeń. Są to:

- **Capabilities** – wyrażenie lub definicja zdolności aplikacji do spełnienia podstawowych funkcji organizacji. Najczęściej jest to główna funkcjonalność aplikacji lub komponentu.
- **Aplikacja (Application)** – dowolny program lub grupa programów zaprojektowana dla użytkownika końcowego.
- **Komponent (Component)** – program lub grupa programów, które mają implementację jednej lub więcej *capabilities*. Najczęściej spotykanymi komponentami są aplikacje, ale ta definicja obejmuje również usługi chmurowe, usługi integracyjne, mikrousługi i inne.
- **Wymaganie (Requirements)** – oczekiwanie co do, jakie biznes definiuje wobec aplikacji i komponentów.

Pomimo że komponent to definicja szersza, w poniższym opracowaniu definicje aplikacji i komponentu traktowane są zamiennie, ponieważ podlegają wszystkim i tym samym procesom architektonicznym. Wykorzystanie słowa „aplikacja” ułatwia odbiór tego opracowania.



1. Wstęp

Tempo, częstotliwość i zakres zmian w otaczającym nas świecie wymuszają zmianę podejścia do zarządzania zmianą. Ponieważ zarządzanie zmianą stało się nową normą, a nie sytuacją nadzwyczajną, oznacza to, że tylko ciągła transformacja firmy pozwoli na jej przetrwanie w obecnych czasach.

Lean Architecture Framework jest zbiorem dobrych praktyk, dzięki którym w sposób spójny i szybki środowisko IT będzie odpowiadać na zmieniającą się sytuację biznesową. Robi to poprzez odpowiednie umieszczenie architekta w procesie wytwarzania oprogramowania. Architekt uzbrojony we właściwą wiedzę potrafi szybko podejmować dobre decyzje.

LAF nie tylko wpisuje się w realizację ciągłej transformacji: on ją także aktywnie wspiera. LAF robi to poprzez ciągłe zmiany w celach architektonicznych i procesie ciągłego dążenia do ich osiągnięcia.

LAF wspiera zaangażowanie wszystkich architektów oraz biznesu, budując porozumienie między nimi dążąc do zbudowania między nimi trwałej, zdrowej relacji. Dzięki temu pozwala na rozwój osobisty architektów pracujących w organizacji, co z kolei ma bezpośredni wpływ na podejmowanie dobrych decyzji dotyczących środowiska IT.

Innowacyjność jest sednem LAF. W pracy dzień po dniu architekt ma możliwość proponowania nowych rozwiązań podczas bezpośredniej współpracy z biznesem. LAF wspiera rozpowszechnianie i realizację pomysłów biznesowych w całej organizacji.

LAF charakteryzuje się takimi cechami:

Simple (Prosty)	LAF jest łatwy do zrozumienia i krótki.
Pragmatic (Pragmatyczny)	LAF koncentruje się tylko na działaniach, które przynoszą wartość i na niczym więcej.
Adaptable (Adaptowalny)	LAF daje się łatwo przystosować do specyfiki i potrzeb organizacji.
Complete (Kompletny)	LAF służy do zarządzania wszystkimi aspektami złożonego środowiska architektury IT, jest to pełnowartościowe rozwiązanie, które zawiera sprawdzone wytyczne.
Enables Architectural Agility (Wspierający Zwinność w Zakresie Architektury)	LAF wspiera ciągłą adaptację ze względu na sytuację organizacji oraz umożliwia dostosowanie się do zmieniającego się otoczenia biznesowego.



2. Dlaczego dobra architektura jest potrzebna

Każde przedsiębiorstwo, aby być elastyczne, otwarte na zmiany, a przy tym efektywne kosztowo, nie w pojedynczych obszarach biznesowych, ale jako cała organizacja, musi zarządzać architekturą IT, a w szczególności środowiskiem aplikacyjnym jako istotną jej częścią.

LAF odpowiada na te potrzeby zapewniając pełne wytyczne dla organizacji, definiując role, procesy i artefakty.

Głównym celem wdrażania LAF w organizacjach jest wspieranie Continuous Transformation poprzez:

- kontrolowanie złożoności środowiska aplikacji IT zapobiegając jego niekontrolowanemu rozrostowi i powielaniu funkcjonalności aplikacji w organizacji;
- wyznaczenie architektonicznej ścieżki umożliwiającej rozwój biznesowy i technologiczny organizacji;
- utrzymanie aktualności stosu aplikacyjnego zapobiegające degradacji technicznej, a w konsekwencji także bankructwu technologicznemu;
- osadzanie celów długoterminowych i misji organizacji w procesie dostarczania oprogramowania zapewniające lepsze zrozumienie ich w całej organizacji i prowadzące w ten sposób do ich realizacji;
- wspieranie elastyczności organizacji w zakresie adaptacji zmian biznesowych jako odpowiedzi na wymagania rynku i konkurencję;
- utrzymanie kosztów na optymalnym poziomie.

LAF uznaje, że dzisiejsze organizacje znajdują się pod ciągłą presją wymagających rynków i ich konkurentów, dlatego muszą być w ciągłej transformacji. Stoi to w opozycji do dużego projektu transformacyjnego, który ledwo nadąża za zmieniającym się rynkiem. Dzięki odpowiednio zarządzanej architekturze IT optymalne pod względem kosztów, spójne środowisko jest gotowe do dalszego wzrostu i zmian, umożliwiając rozwój biznesu.

Każda organizacja ma własny proces dostarczania oprogramowania SDLC oparty na zwinnym podejściu, takim jak Scrum, XP, Less, SAFe itp. lub bardziej tradycyjnym, takim jak waterfall. Wszystkie potrzebują ram architektonicznych, aby zachować spójność i gotowość do rozwoju.

3. Elementy LAF

LAF składa się z trzech głównych części:

- **DEFINITION** – jest to część, w której architekci wraz z innymi osobami w organizacji tworzą produkty będące podstawą do realizacji architektury:
 - **REFERENCE ARCHITECTURE (RA)** – oczekiwany kształt portfela aplikacji, który ma wspierać zmieniający się biznes;
 - **STANDARDS (ST)** – standardy pozwalające utrzymać środowisko aplikacyjne w stanie pozwalającym na przewidywalne zarządzanie;
 - **COMPONENT PASSPORT DATABASE (CPD)** – aktualny stan portfela aplikacji ułatwiający podejmowanie decyzji o tym, czy komponent nadaje się do rozwoju.

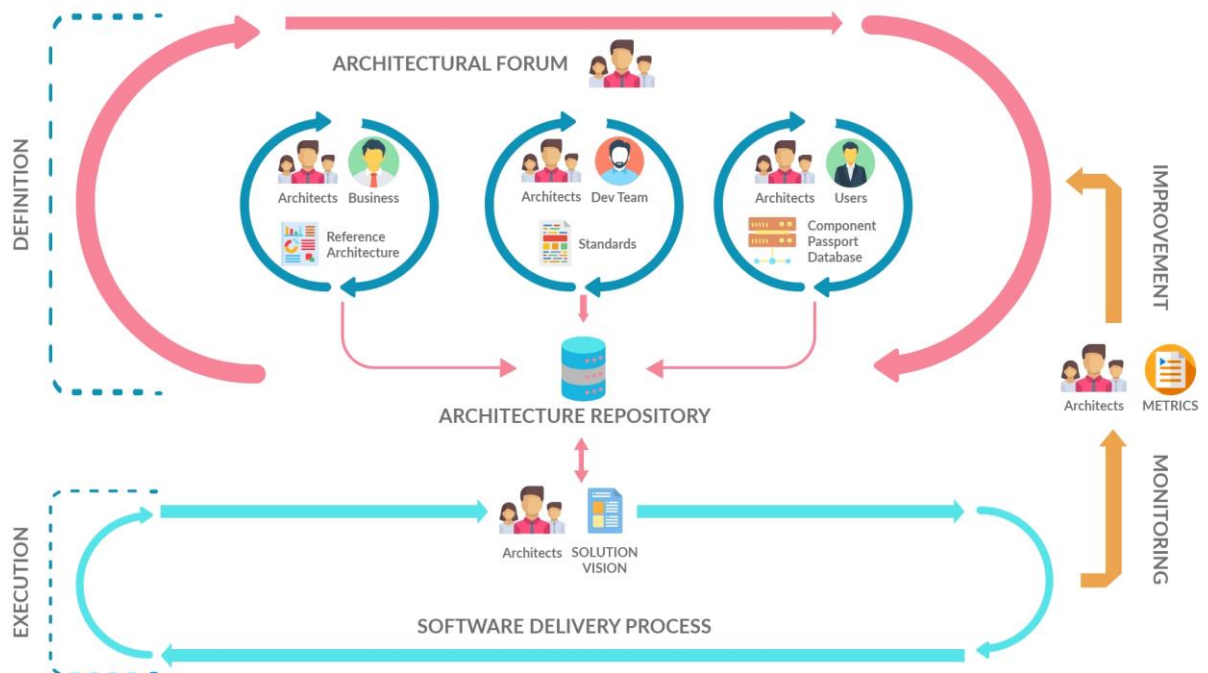


- **EXECUTION** – w tej części architektura jest implementowana poprzez udział architektów w procesie budowy oprogramowania (SDLC). Architekci opisują wizję rozwiązania odpowiadającą na wymagania biznesu i jednocześnie ich zgodność z wytycznymi dla architektury.
- **IMPROVEMENT** and **MONITORING** to najmniejsza część LAF mająca istotny wpływ na całość architektury w organizacji. Jej zadaniem jest obserwacja jakości architektury i procesów architektonicznych, a w kolejnym kroku usprawnienia w tych obszarach.

Częścią wspólną dla wszystkich elementów jest **ARCHITECTURE REPOSITORY (REPO)** – baza, w której umieszczane są poszczególne produkty. Stanowi ona także źródło informacji do podejmowania decyzji przy tworzeniu wizji rozwiązania.

Całością architektury zarządza i buduje grupa architektów zebranych w formalny zespół, który odbywa cykliczne spotkania zwane **ARCHITECTURAL FORUM (FORUM)**. Efektywna praca tego zespołu jest najważniejszym czynnikiem wpływającym na jakość architektury. W związku z tym LAF wprowadza też praktyki związane z organizacją pracy.

Całość LAF została zobrazowana na poniższym diagramie.



Lean Architecture Framework Footprint

3.1. Artefakty

3.1.1. Reference Architecture

Reference Architecture to długoterminowy, wysokopoziomowa wizja architektoniczna dla danego obszaru biznesowego, w oparciu o którą wszystkich dalszych rozwiązań (Solution



Vision) będą tworzone. Jest to kierunek i wytyczne dla decyzji architektonicznych, stan „TO BE”, który najprawdopodobniej nigdy nie zostanie osiągnięty. Wynika to ze zmieniającego się biznesu, a co za tym idzie, zmieniającej się architektury referencyjnej. RA może być zmieniona przed jej realizacją lub osiągnięciem. W pełni udostępnia potencjał biznesowy, realizując jego cele i potrzeby zapewnienia zgodności z misją i strategią firmy. Ustanawia podstawę techniczną dla dalszego rozwoju organizacji. Suma wszystkich architektur referencyjnych tworzy architekturę referencyjną dla całej organizacji.

Jest to lekka architektoniczna wizja pożądanego stanu środowiska aplikacji, do którego będzie się dążyć w dłuższej perspektywie. Zwykle zawiera schemat z komponentami, *capabilities* i interakcjami między nimi. Powinna również zawierać krótki opis każdego komponentu oraz korzyści wynikających z jej implementacji. Podczas jej tworzenia nie są przyjmowane żadne założenia dotyczące zasobów. Nie obejmuje żadnego stanu pośredniego ani informacji o tym, jak zostanie to osiągnięte.

Jest tworzony w ścisłej współpracy z jednostką biznesową i innymi zainteresowanymi stronami, w pierwszej kolejności odpowiada na ich potrzeby. Dlatego ważne jest, aby była utworzona za pomocą łatwo zrozumiałego języka oraz posiadała spójną formę. AR uwzględnia trendy biznesowe, a także trendy technologiczne i może stworzona w odniesieniu do architektury branżowej, jeśli taka jest dostępna. Uwzględnia pełny potencjał biznesowy, realizując jego cele i potrzeby, zapewniając zgodność z misją i strategią firmy. Stanowi techniczną podstawę dla dalszego rozwoju organizacji. Agregacja wszystkich architektur referencyjnych tworzy widok architektury całej organizacji.

REFERENCE ARCHITECTURE jako zbiór *capabilities* środowiska aplikacyjnego stanowi cel dla zmian architektonicznych. Każde wdrożenie aplikacji powinno zbliżać całość środowiska do osiągnięcia tego celu. RA ulega zmianom ze względu na zmiany rynkowe i wewnętrzne organizacji. Ciągłe zmiany w dążeniu do zmieniającego się celu są sednem *Continuous Transformation*. Całość została zobrazowana na poniższym diagramie.

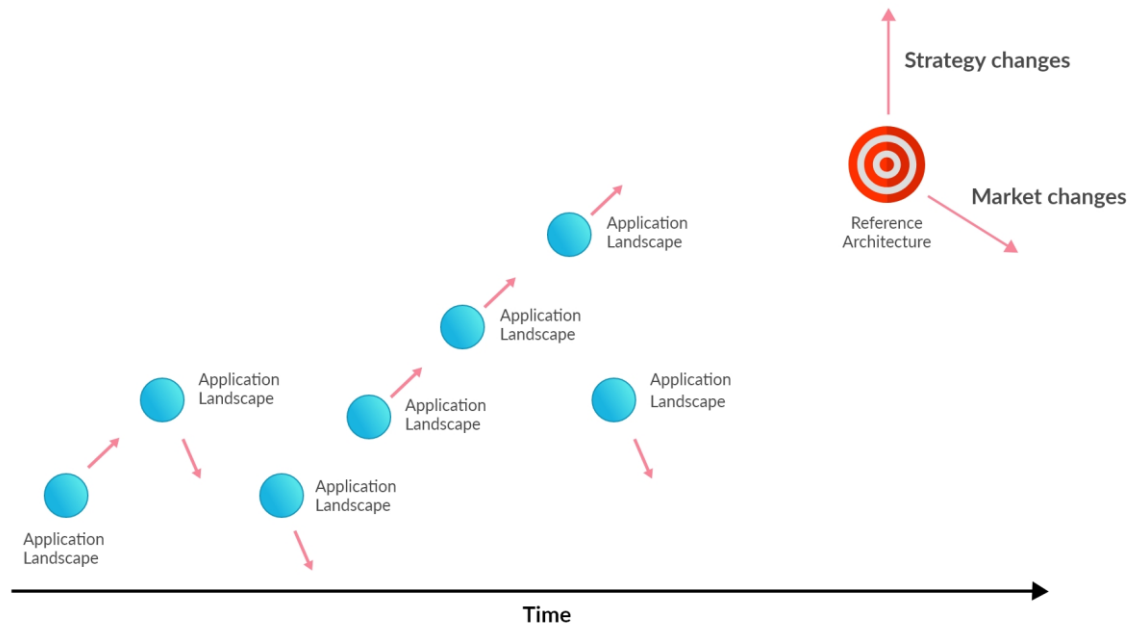


Diagram pokazujący w abstrakcyjny sposób zmiany środowiska aplikacyjnego w kierunku architektury referencyjnej

Proces tworzenia REFERENCE ARCHITECTURE zwykle koncentruje się na konkretnej jednostce biznesowej. Architekt jest odpowiedzialny za tworzenie RA w ścisłej współpracy z jednostką biznesową odpowiadając na ich potrzeby w pierwszej kolejności poprzez, organizację serii warsztatów, aby dogłębnie zrozumieć potrzeby dziedziny biznesowej.

Podczas tworzenia RA architekci mają możliwość a nawet więcej - są zobowiązani do proponowania innowacyjnych rozwiązań i wsparcia biznesu przy wdrażaniu nowoczesnych rozwiązań.

Tworzenie i utrzymywanie architektury referencyjnej to ciągła praca, która może zależeć od różnych czynników, takich jak nauka płynąca z jej implementacji, zmiany strategii biznesowych i technologicznych, zmiany warstwy technologicznej lub przepisów prawnych lub presji rynku itp. Dlatego RA powinna być dostosowywana w zależności od wystąpienia powyższych czynników, niezależnie od ich wystąpienia RA powinna zostać poddana przeglądowi oraz aktualizacji nie rzadziej niż raz w roku celem zachowania aktualności.

3.1.2. Standardy

Wydajne i przewidywalne zarządzanie środowiskiem IT wymaga, aby architektura aplikacji była spójna na wielu płaszczyznach. Utrzymanie ujednoliconych elementów w środowisku aplikacyjnym pozwala nie tylko ograniczyć chaos w architekturze, ale też przyspieszyć podejmowanie decyzji w obszarach objętych standaryzacją. Standardy tworzą bezpieczną przestrzeń do eksperymentów dla zespołów zaangażowanych w rozwój i utrzymanie oprogramowania.



Narzędziem utrzymania tego porządku są formalne artefakty, np. dokumenty czy strony Wiki opisujące te standardy, ograniczenia i wymagane modele realizacji funkcjonalności w środowisku IT.

Standaryzacja ta powinna być zrealizowana na minimalnym potrzebnym poziomie i ograniczać się do najważniejszych wymagań. Celem jest znalezienie takiego zestawu standardów, który wytworzy spójne środowisko, a jednocześnie nie ograniczy innowacyjności.

Najważniejsze typy standardów które należy opracować to:

- spójność technologii – ograniczenie ilości technologii w organizacji do zdefiniowanej listy i utrzymanie spójności co do wersji wszystkich jej elementów,
- spójność integracji – ujednoczenie i zdefiniowanie dobrych praktyk integracji międzysystemowej,
- spójność monitorowania i logowania – ujednoczenie procesu logowania i monitorowania zdarzeń systemowych i biznesowych w środowisku aplikacyjnym,
- spójność w zarządzaniu danymi – ujednoczenie sposobu zarządzania danymi w zakresie budowy modeli, ich spójności i odpowiedzialności za to systemów.

Proces tworzenia

Budowa i uaktualnianie standardów powinno być oparte na cyklicznym procesie, np. rocznym, w którym zespół złożony z architektów przygotowuje treść standardu, później uzgadnianą z zainteresowanymi stronami. Na przykład szczegóły standardu technologicznego, dotyczące używanego języka programowania, powinny być ustalone z zespołami deweloperskimi, które używają tej technologii. Standardy powinny być dostępne i komunikowane wszystkim zainteresowanym stronom łącznie z dostawcami.

2.1.3. Component Passport Database

Kolejny bardzo istotny artefakt to baza danych zawierająca podstawowe informacje o komponentach, zebrane w trzech podstawowych kategoriach: biznesowej, technicznej i finansowej. Informacje te zbierane są w celu ustalenia, jaka powinna być przyszłość komponentu.

W perspektywie biznesowej aplikacja oceniana jest pod kątem znaczenia w procesach biznesowych. W kategorii technicznej oceniana jest jakość kodu, jakość procesów dostarczania i zgodność technologii ze standardami.

Perspektywa finansowa to ocena kosztów komponentu w zakresie licencji, infrastruktury, utrzymania i kosztów rozwoju. Im niższe koszty, tym komponent jest lepiej oceniany. Kontekst finansowy oceny powinien oprzeć się na uśrednionych wartościach występujących w organizacji. Przy obliczaniu oceny wynik może być oparty na przykład na 5-stopniowej skali, niekoniecznie na rzeczywistych wartościach.

Ostatecznie każda z aplikacji powinna trafić do jednej z czterech grup przeznaczenia:

- inwestycja – komponent oceniony technicznie bardzo dobrze i mający duże znaczenie biznesowe. Komponenty te przeznaczone są do rozwoju.



- migracja – komponent oceniony technicznie źle i mający duże znaczenie biznesowe. W komponentach tych niezbędne jest przeniesienie funkcjonalności do innych aplikacji.
- tolerowanie – komponent oceniony technicznie bardzo dobrze, ale mający małe znaczenie biznesowe. Komponenty te ze względu na niski koszt i dobry stan techniczny utrzymuje się w środowisku.
- eliminacja – komponenty w złym stanie technicznym, z funkcjonalnościami, które są nieistotne lub pokryte przez inne aplikacje, powinny być przeznaczone do wygaszenia.

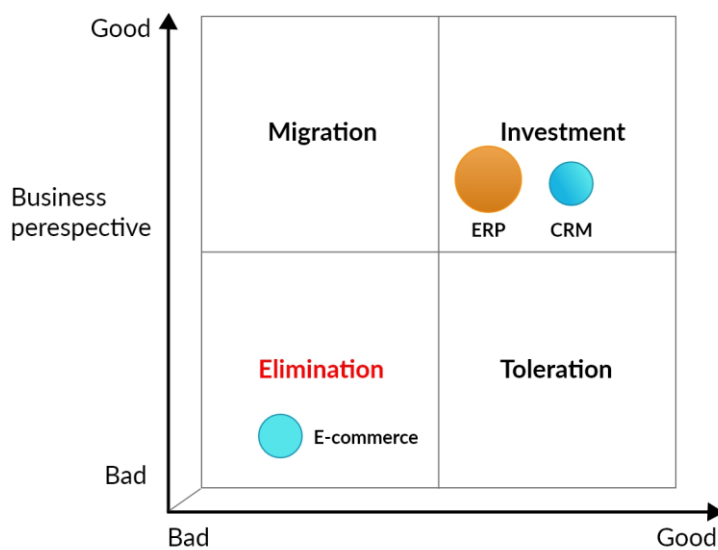


Diagram prezentujący sposób przydziału komponentów do portfeli. Wielkość punktu obrazuje perspektywę finansową każdego z komponentów (im większy punkt, tym aplikacja otrzymuje lepszy wynik)

Przydział do kategorii jest istotny w przypadku porównania, w której z dwóch aplikacji powinien nastąpić rozwój, jeśli dwie aplikacje mają redundantne funkcjonalności.

Proces

W procesie uzupełniania bazy, który powinien być cykliczny, zaleca się, by jednym ze składników oceny była ocena użytkowników końcowych uzyskana poprzez ankiety oceniające zadowolenie z pracy z daną aplikacją. Dobrą praktyką jest zastosowanie Net Promoter Score (NPS) w ich ocenie.

Ostateczna ocena krytycznych biznesowo aplikacji powinna być szczegółowo omówiona, uzgodniona i zaakceptowana przez stronę biznesową. Zespół techniczny powinien zadbać o regularną aktualizację komponentów

Cechy techniczne komponentu powinny być na bieżąco uzupełniane przez zespół deweloperski.

Proces oceny powinien zawierać następujące kolejne kroki:



- ocena techniczna aplikacji przez architekta,
- ocena użytkownika aplikacji,
- ocena finansowa,
- ocena biznesowa aplikacji przez właściciela biznesowego,
- kategoryzacja aplikacji,
- komunikacja o zmianie przypisania aplikacji do kategorii, jeśli dotyczy

2.1.4. Solution Vision

Solution Vision (SV) wysokopoziomowy projekt architektoniczny, który odpowiada na bieżące potrzeby biznesowe. Zmiany w warstwie architektonicznej są ich integralną częścią. Solution Vision jest zawsze powiązany z wartością biznesową, ponieważ zmiany w architekturze nigdy nie są celem samym w sobie. Solution Vision krok po kroku wprowadza zmiany dążąc do architektury referencyjnej. To prawdziwe serce stopniowej, ciągłej transformacji organizacji w zakresie architektury aplikacji. W niektórych przypadkach może zawierać zmiany, takie jak wdrożenie nowego komponentu lub aplikacji, przeniesienie Capabilities z jednego komponentu do drugiego, wycofywanie aplikacji lub komponentu, technologii lub Capabilities, migracje itp.

Jednym z najważniejszych elementów dobrego SV jest zaangażowanie programistów w jego tworzenie. Skuteczna współpraca architekta z programistami / devopsami / administratorami jest kluczowa dla zapewnienia wysokiej jakości architektury i jej efektywnego wdrożenia. Zrozumienie założeń przedstawionych przez architekta powinno być budowane poprzez udział architekta w codziennym życiu zespołu deweloperskiego.

W kontekście budowania SV architekt wraz z biznesem jest odpowiedzialny za tworzenie wysokiej jakości wymagań niefunkcyjnych (NFR), w tym wymagań bezpieczeństwa. Architekt powinien mieć pewność, że budowane przez niego rozwiązanie bazuje na dobrym NFR.

Proces

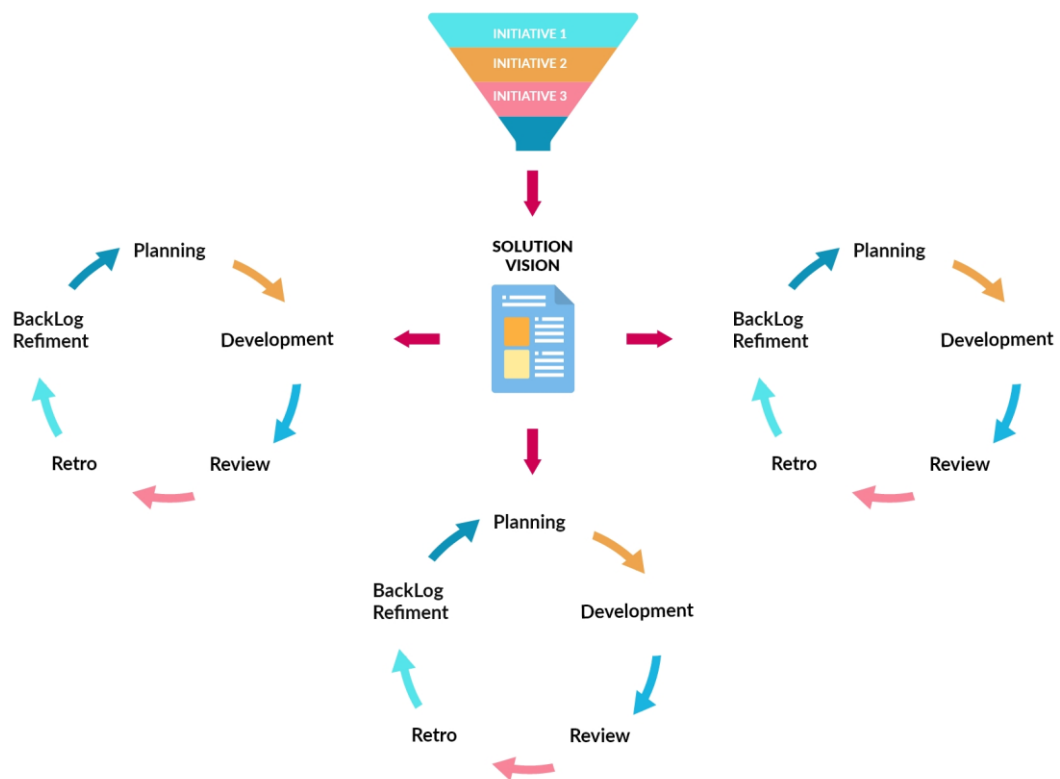
Tworzenie Solution Vision jest wbudowane w SDLC organizacji i zawsze jest wyzwalane przez bieżące wymagania biznesowe. Przekształca inicjatywę biznesową w projekt architektoniczny jednej lub większej liczby aplikacji, pokazując je jako logiczne komponenty z ich *capabilities* i interakcją między nimi. Powinien obejmować wszelkie zmiany w warstwie sprzętowej i określać wymagania niefunkcyjne dla inicjatywy biznesowej, które są istotnym elementem dla dobrego projektu architektonicznego.

Kluczem do prawidłowego wdrożenia tego procesu jest dokonywanie stopniowych zmian w warstwie architektonicznej, wprowadzając stopniowo nowe komponenty, interakcje między nimi i *capabilities* w pełnej zgodności z Reference Architecture, informacją z CPD i standardami architektonicznymi. Dzięki takiemu podejściu organizacje osiągają szybciej korzyści oraz mają szansę sprawdzić korzyści, które były początkowo zakładane, dostając możliwość szybszej adaptacji i uczenia się z ich implementacji, oszczędzając pieniądze w porównaniu z dużym projektem transformacyjnym (projekty typu big bang).



Celem przedsiębiorstwa nie jest wprowadzanie zmian w warstwie architektury samych w sobie, ale wprowadzanie tych zmian, które przynoszą wartość dla organizacji. Dlatego w organizacjach powinniśmy wprowadzać tylko tyle zmian, które zapewniają właściwy wynik ekonomiczny i technologiczny rozwoju przedsiębiorstwa. Proces architektoniczny musi być zatem ściśle połączony z procesem dostarczania oprogramowania dla przedsiębiorstw.

Na przykład SV może być wprowadzony w środowisko wytwarzania oprogramowania oparte o metodykę SCRUM w modelu, w którym dokument ten jest daną wejściową dla poszczególnych zespołów. Architekt buduje wizję całości rozwiązania dla całej inicjatywy i jest ona przekazywana poszczególnym zespołom rozwijających aplikacje. Zostało to zobrazowane na poniższym diagramie.

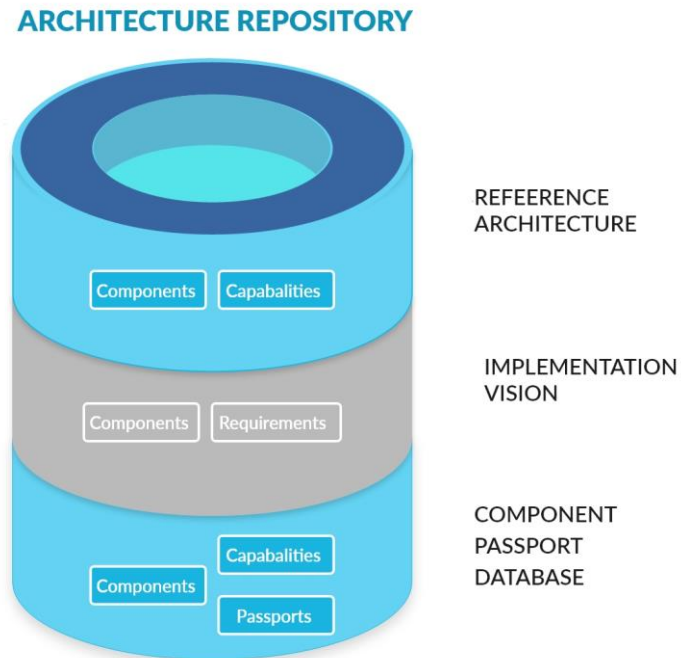


Abstrakcyjne zobrazowanie umieszczenia Solution Vision w procesie dostarczania oprogramowania

2.1.5. Repository

Produkty poszczególnych procesów powinny znaleźć się w jednym centralnym repozytorium architektury. Baza ta powinna zawierać informacje o aplikacji w kontekście:

- przyszłości – miejsce w RA wraz z oczekiwanymi *capabilities*,
- *capabilities* jakich dostarcza – zespół wymagań i opracowań SV związanych z tym,
- jej aktualnego stanu – cechy aplikacji oraz zebrane w postaci paszportu.



Ref. Reprezentacja graficzna REPOSITORY

Kształt repozytorium, w tym lista cech aplikacji, sposób prezentacji *capabilities* czy sposób zapisu SV zależy od organizacji i powinien być zoptymalizowany w procesie udoskonalania. Repozytorium powinno być zbudowane w sposób pozwalający na dokonywanie w nim zmian. Zawartość repozytorium powinna stanowić minimalny zestaw danych potrzebny do klasyfikacji komponentu lub przedstawienia architektury referencyjnej. Jest to niezbędne ze względu na potrzebę posiadania aktualnych danych w REPO. Zbyt wysoki koszt ich pozyskania będzie miał ogromny wpływ na ich jakość.

2.1.6. Metryki

Podczas gdy prawdziwa transformacja w kierunku architektury referencyjnej odbywa się w modelu przyrostowym przez „Solution Vision”, potrzebne są mierniki, które zapewnią, że organizacja pozostanie na właściwej drodze. Proponowane wskaźniki są dobrą bazą do monitorowania tego postępu oraz doskonałym źródłem do retrospekcji dla organizacji i architektów. Zazwyczaj dane te są zbierane co kwartał; czas ten może być dostosowywany w zależności od potrzeb organizacji.

Monitoring wskaźników architektonicznych umożliwi nam:

- Uzyskanie informacji czy organizacja naprawdę się zmienia w warstwie architektonicznej
- Uzyskanie wiedzy jak decyzje biznesowe wpływają na architekturę
- Podejmowanie świadomych decyzji dotyczących kompromisów między szybkością, wartością biznesową i zmianami architektury



Każdy z proponowanych wskaźników powinien mieć ustalony poziom bazowy i oczekiwany w celu lepszego monitorowania postępu, a każde naruszenie ustalonego poziomu bazowego lub oczekiwanego powinno prowadzić do otwartej dyskusji, retrospekcji i dostosowań.

Reference Implementation Metric (RIM) – każdy projekt implementacji (SV) powinien być dostosowany i zgodny ze standardami oraz oceną w CPD i zmierzać w kierunku spełnienia AR, ale w niektórych przypadkach nie można tego osiągnąć ze względu na krytyczne czasowo projekty, regulacje prawne itp. Dlatego każdy SV powinien zawierać atrybuty określające jego całkowitą zgodność z architekturą referencyjną. Zwykle ten atrybut może przyjąć jedną z trzech wartości:

- **„Zgodny”** – oznacza, że projekt implementacji jest w pełni kompatybilny z architekturą referencyjną, a jego implementacja będzie stopniowo przekształcać krajobraz aplikacji do pożądanego stanu, zaprojektowanego w architekturze referencyjnej, szczególnie pod względem używanych komponentów, ich możliwości i interakcji między nimi.
- **„Neutralny”** – jego wdrożenie nie wpływa na krajobraz aplikacji, szczególnie pod względem standardów, używanych komponentów, interakcji między nimi i ich możliwości. Zwykle jest to niewielka zmiana lub zmiana UX.
- **„Niezgodny”** – występuje wtedy, gdy projekt implementacji nie może spełnić wymogu zgodności z architekturą referencyjną. Taka sytuacja ma miejsce szczególnie wtedy, gdy niektóre starsze komponenty są używane i aktualizowane, jest wykorzystywana niechciana technologia, dodane są nowe komponenty lub dodatkowe interakcje, albo stwierdzono jakiegokolwiek inne naruszenia architektury referencyjnej.

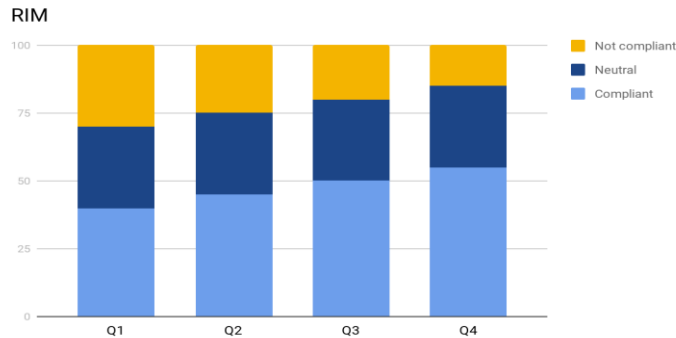
Tak może wyglądać wzór na obliczenie wskaźnika:

$$RIM = \frac{CV}{AV}$$

CV= Liczba Solution Vision zgodnych z architekturą ref i standardami

AV= Liczba wszystkich Solution Vision

RIM jest jednym z kluczowych wskaźników używanych do śledzenia kierunków zmian, określanym jako procent każdej wartości atrybutu w określonym czasie, np. raz na kwartał. Z poniższego wykresu można odczytać trend opracowanych zmian. Na tej podstawie organizacja może dostosować tempo transformacji inwestując więcej czasu i środków w „zgodne” projekty wdrożeniowe.



Przykład raportowania stanu metryki RIM

Application Reference Metric (ARM) – to wskaźnik aplikacji używanych w bieżącym środowisku aplikacyjnym organizacji do aplikacji w CPD o statusie „Inwestycja” i „Tolerancja”.

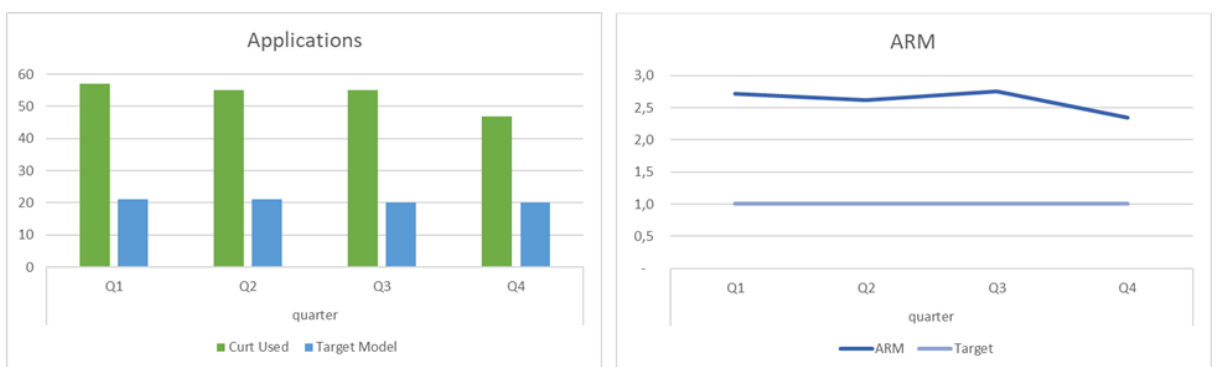
Dzięki zastosowaniu tej metryki jesteśmy w stanie sprawdzić jakość komponentu w naszym portfolio. Możemy sprawdzić, czy powstają aplikacje warte inwestycji, a te słabe biznesowo i technologicznie są eliminowane.

$$ARM = \frac{AP}{IA}$$

AP = Liczba wszystkich komponentów w CPD

IA = Liczba komponentów w CPD w statusie Inwestycja i Tolerancja

Na przykład, jeśli wszystkich aplikacji jest w środowisku 100, a tych o statusie „Inwestycja” i „Tolerancja” 50, to wskaźnik ARM wynosi 2. W dłuższej perspektywie wskaźnik ten powinien dążyć do wartości 1.



Przykład raportowania stanu metryki ARM

Implemented Capabilities Redundancy Metric (ICRM) – jest to miara nadmiarowości zaimplementowanych *capabilities*. Miara jest liczba wystąpień *capabilities* w istniejących aplikacjach w stosunku do docelowej liczby *capabilities*.



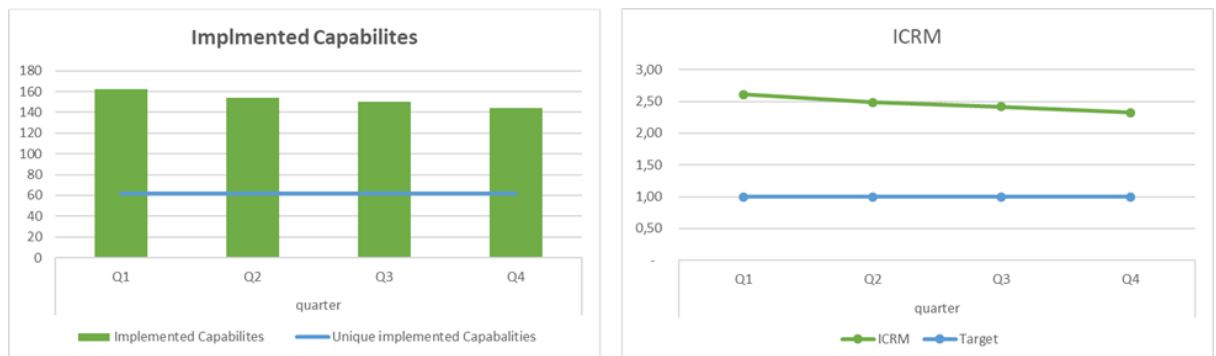
Ta miara służy do monitorowania czy funkcje nie są budowane nadmiarowo powyżej oczekiwanego poziomu. Wartość docelowa nie zawsze powinna wynosić jeden. Czasami warto mieć kilka instancji *capabilities*.

$$ICRM = \frac{EC}{TC}$$

EC= Liczba instancji wszystkich *capabilities*

TC= Oczekiwana liczba instancji *capabilities* we wdrożonych aplikacjach

Przykładowo, jeśli w naszym uproszczonym środowisku jest zaimplementowanych 6 *capabilities* „send SMS to Customer”, a powinny być tylko 2, to współczynnik ICRM dla takiego teoretycznego środowiska wynosi 3. Wartość współczynnika powinna dążyć do 1.



Przykład raportowania stanu metryki ICRM

Capability Reference Metric (CRM) - jest to stosunek *capabilities* wykorzystanych w bieżącym środowisku aplikacyjnym organizacji do tych zaproponowanych w architekturze referencyjnej (RA) organizacji.

Ta miara służy do monitorowania, czy organizacja dąży do spodziewanego zbioru *capabilities*. Możemy regularnie sprawdzać, czy projekty / inicjatywy dodają nowe *capabilities*

$$CRM = \frac{UC}{TC}$$

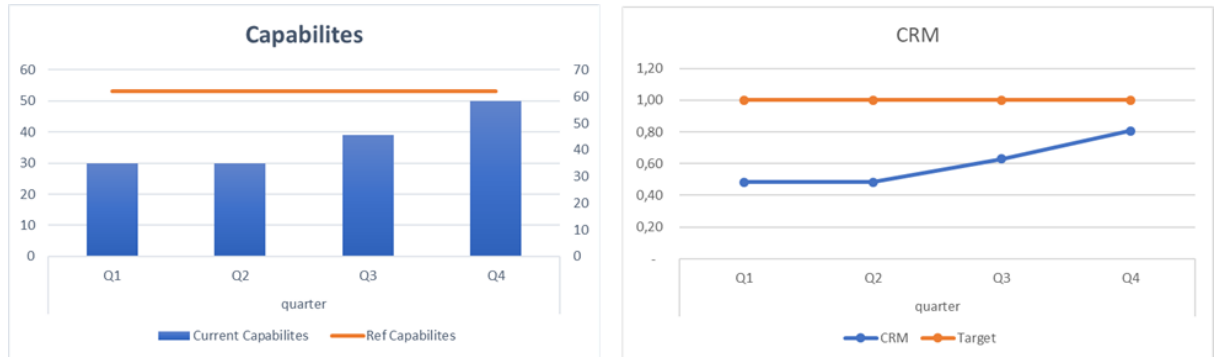
UC= Liczba typów *capabilities* będących aktualnie w środowisku

TC= Liczba *capabilities* proponowane w architekturze referencyjnej organizacji (AR)

Przykładowo, jeśli w organizacji jest zaimplementowanych 46 *capabilities*, a w RA jest ich 100, to współczynnik CRM wynosi 0,46. Metryka powinna być mierzona w określonym



interwale czasowym, np. raz na kwartał. W dłuższej perspektywie liczba posiadanych *capabilities* powinna dążyć do poziomu z architektury referencyjnej, a wskaźnik do poziomu 1.



Przykład raportowania stanu metryki CRM

2.2. ARCHITECTURAL FORUM

Dla efektywnej realizacji architektury LAF wymaga doskonałej komunikacji pomiędzy wszystkimi architektami. Aby pozostać niezależni, muszą oni poznać wpływ swojej pracy na innych architektów i ich obszary, a także zrozumieć całościowy obraz przedsiębiorstwa, dlatego LAF wprowadza Forum Architektoniczne jako główną platformę komunikacji. Spotkanie to powinno odbywać się co najmniej raz w tygodniu i trwać od 2 do 4 godzin. Są to nieformalne spotkania dla architektów w celu przedstawienia dokonanych przez nich głównych odkryć, uzgodnienia używanego języka i kształtu repozytorium architektury, omówienia wyników architektury referencyjnej, uzgodnienia nowych komponentów i nowych *capabilities* wprowadzonych do organizacji. Architekci mają okazję do przedstawienia i omówienia swoich wątpliwości w odniesieniu do „projektów wdrożeniowych”. W trakcie spotkań należy omawiać wprowadzenie i zmodyfikowanie standardów stosowanych w organizacji oraz wszelkie inne ważne kwestie, które pojawią się w procesach realizacyjnych.

Spotkanie to służy również jako sesja planowania zadań architektonicznych, takich jak utrzymanie repozytorium, przeprowadzenie procesu oceny komponentów, zarządzanie obszarami objętymi procesem architektury referencyjnej, odświeżenie standardów przedsiębiorstwa lub wprowadzenie nowych.

Forum architektoniczne nie zostało stworzone w celu zaplanowania prac związanych z „projektowaniem implementacji”, ponieważ jest to zadaniem SDLC. To spotkanie prowadzi do lepszego zrozumienia całej organizacji przez poszczególnych architektów i ponownego wykorzystania komponentów, możliwości, usług oraz technologii.

Zespół

The Architects Team (zespół architektów) ma płaską strukturę, składa się z ludzi z głęboką wiedzą technologiczną, a także z głębokim zrozumieniem kontekstu biznesowego, architekci powinni i mogą pełnić różne role związane z podziałem technologicznym i biznesowym. Jest to samoorganizujący się zespół, który optymalizuje swoją wydajność i zapewnia wysoki poziom komunikacji, ponieważ jest on kluczowym czynnikiem sukcesu. Członkowie zespołu mają dużą autonomię i pełną odpowiedzialność za dostarczanie artefaktów LAF. Dzielą te same normy, zasady i standardy. Są wspierani przez Master Architect, jest to rola



odpowiedzialny za zapewnienie, że zespół architektoniczny postępuje zgodnie z procesem i praktykami LAF. Wspiera współpracę między członkami zespołu, zapewniając otwarte środowisko do otwartej dyskusji, prowadząc do wymiany wiedzy i obaw, pomaga usuwać przeszkody i zapewniać odpowiednią jakość procesów architektonicznych i repozytorium. Odgrywa specjalną rolę w sytuacjach trudnych, takich jak brak komunikacji w zespole lub niemożność znalezienia szybkiego kompromisu, jaki często jest potrzebny. Rola Master Architect jest analogiczna do roli Scrum Mastera

W dużych organizacjach, w których liczba architektów jest większa niż 8, dobrze jest wydzielić mniejsze zespoły skoncentrowane na jakimś obszarze biznesowym. Te zespoły mają własne spotkania architektoniczne i sesje doskonalenia. Każdy zespół powinien wybrać swoich przedstawicieli, w tym Mistrza Architekta, do udziału w spotkaniu poświęconym całej architekturze. Przy takim podziale warto wyznaczyć Lead Master Architect odpowiedzialnego za współpracę zespołów architektonicznych.

Wielkość zespołu architektonicznego zależy od wielkości organizacji, jej złożoności i dojrzałości procesu dostarczania oprogramowania. Celem jest osiągnięcie minimalnego składu, który zapewni płynną realizację procesu dostarczania oprogramowania.

4. Wdrożenie LAF

4.1. Najlepsze praktyki

4.1.1 Wspólny język

Jednym z najważniejszych czynników sukcesu wdrożenia LAF, a co za tym idzie, uzyskania dobrej jakości architektury, jest wypracowanie wraz z biznesem i programistami wspólnego języka. Odpowiednio zbudowany język pozwala na prowadzenie dyskusji o architekturze przedstawicieli biznesu między sobą lub z ludźmi z działu IT.

Język ten powinien być prosty i opierać się na prostej definicji tego, czym są komponent i *capabilities* oraz w jaki sposób określamy przyszłość danego obszaru biznesowego.

Zalecamy stosowanie koncepcji DD (projektowanie oparte na domenie) jako metody tworzenia odpowiedniego języka komunikacji

4.1.2 Podział

Większość dużych organizacji ma złożony model biznesowy, oparty na mnogości produktów, usług lub procesów. W takich organizacjach warto, by architektura referencyjna była przygotowywana dla danego obszaru biznesowego. Całość architektury referencyjnej należy podzielić, najlepiej zgodnie z podziałem struktury organizacyjnej lub na różne strumienie wartości (*value stream*), i każdą z części opracowywać oddzielnie. W ten sposób powstają dla danej domeny biznesowe architektury referencyjne.

4.1.3 Architecture Log



W związku z tym, że organizacja się zmienia, a w szczególności następuje rotacja pracowników, bardzo ważne jest, żeby istniała baza wiedzy o uzasadnieniu podjętych decyzjach. Pozwala to na utrzymanie ciągłości działań i konsekwentne dążenie do celu. Architecture Log jest bazą, w której powinny być przechowywane między innymi tak ważne decyzje jak:

- decyzja o wyborze zakupu nowej technologii lub aplikacji,
- akceptacja wyjątku od Architektury Referencyjnej lub Standardu,
- publikacja AR lub standardów,
- eliminacja aplikacji ze środowiska.

4.1.4 Continuous improvement sessions

Co 2-4 tygodnie powinna odbywać się sesja ciągłego doskonalenia dla architektów. Spotkanie powinno trwać od 1,5 godziny do 3 godzin. Sesja retrospekcji jest dobrze znana w Agile Ways Of Working i bardzo dobrze opisana w publicznie dostępnych materiałach, dlatego właśnie konkretna technika powinna zostać zaadoptowana z tego świata i nie będzie omawiana w tym materiale. Jeśli organizacja stosuje już zwinne podejście, zaleca się sięgnięcie po techniki dobrze jej znane. Ważne jest, aby zauważyć, że architekci powinni brać udział w retrospekcjach dotyczących dostarczania oprogramowania niezależnie od Architektonicznych sesji ciągłego doskonalenia wnosząc wszelkie globalne wnioski na to spotkanie.

Przykładowym wynikiem sesji może być: zmiana w modelu oceny komponentów, zmiana w szablonie SV.

4.1.5 Szybka ścieżka

Nie każda inicjatywa powinna być realizowana w procesie z tworzeniem SV. Inicjatywy w ramach jednej istniejącej aplikacji, które nie zmieniają jej *capabilities*, powinny być realizowane szybką ścieżką.

4.2. Antywzorce

4.2.1 Architektura docelowa

Głównym problemem prowadzącym do nieudanej implementacji LAF jest niezrozumienie, czym jest Architektura Referencyjna. Bardzo często zostaje ona zinterpretowana jako duża wizja transformacyjnej, nazywana programem wraz z harmonogramem, szacowaniem budżetu i innymi elementami projektowymi. Staje się celem bardzo odległym, w większości wieloletnim. Pociąga to za sobą wszystkie konsekwencje projektowania środowiska aplikacyjnego w zmieniającym się równolegle biznesie. W większości przypadków program transformacji jest porzucany przy pierwszym dużym problemie w procesie dostarczania. Wtedy też porzucana jest cała architektura. LAF wspiera stopniowe dążenie do architektury referencyjnej, która zmienia się wraz ze zmieniającą się sytuacją biznesową.



4.2.2 Projektowanie aplikacji

Architekci niebędący deweloperami danej aplikacji i niepracujący bezpośrednio przy implementacji nie powinni projektować implementacji funkcjonalności w danej aplikacji. Za projekt techniczny odpowiedzialne są zespoły, które poszczególne funkcjonalności uzgadniają z biznesem i w sposób odpowiedni dla nich je implementują. Architekci powinni zbudować podstawy do realizacji funkcjonalności w postaci wytycznych co do API, sposobu integracji i dużych zmian technologicznych. Najbardziej efektywnym modelem wdrażania architektury jest udział architektów w życiu zespołów programistycznych. Udział ten zależy od rodzaju projektu i jego etapu. Na początku projektu zalecamy ścisłą współpracę Architekta z zespołem deweloperskim. Na późniejszych etapach Architekt powinien w razie potrzeby wspierać zespół.

4.2.3 Architektura poza procesem dostarczania zmian biznesowych

Jeśli architekci nie są umieszczeni w procesie dostarczania oprogramowania i nie pracują z biznesem ani deweloperami, to w konsekwencji izolowani są od wiedzy, która wpływa na środowisko aplikacyjne i od wiedzy o stanie tego środowiska. W praktyce nie jest wtedy możliwe śledzenie ani realizacja architektury referencyjnej. Zespoły aplikacyjne będą kształtować środowisko w sposób odpowiedni z punktu widzenia aplikacji, którą tworzą, nie widząc szerszego kontekstu. Z kolei biznes pracujący z konkretnymi wymaganiami skupi się na realizacji celów krótkoterminowych. Architekci muszą pracować w procesie rozwoju aplikacji, by móc je kreować i nadzorować zmiany.

4.2.4 Architektura jako wąskie gardło

Zbyt obszerne dokumentowanie procesów i elementów, a w konsekwencji ogromny nakład pracy architektów spowoduje, że architektura stanie się wąskim gardłem. Bardzo szybko następstwem tego faktu będzie realizacja projektów z pominięciem aspektów architektonicznych. Wszystkie repozytoria powinny być budowane na poziomie minimalnego spełnienia wymagań i zawierać tylko informacje potrzebne interesariuszom procesów architektonicznych oraz SDLC

4.3. Narzędzia

LAF jest zbiorem dobrych praktyk niezależnym od narzędzi lub ich braku. LAF można zaimplementować, używając jedynie Excela. Bardzo dobrym pomysłem jest jednak automatyzacja niektórych procesów i przechowanie danych w repozytoriach posiadających większe możliwości analizy. Rekomendujemy budowę i posiadanie:

- repozytorium w postaci centralnej bazy danych dającego możliwość pracy wielu osobom, w tym zespołom aplikacyjnym;
- portalu architektury zawierającego informacje o standardach, architecture log i architekturze referencyjnej;
- aplikacji automatyzującej zbieranie ankiet od użytkowników końcowych aplikacji;
- narzędzia do automatyzacji zbierania informacji o aplikacji, np. z systemów statycznej analizy kodu czy bug trackerów;
- narzędzi do automatyzacji zbierania metryk.



Załącznik. Zmiany między wersjami 1.1 i 1.2

Zmiany związane z opiniami z wdrożeń:

- Poprawa definicji metryki i dodawanie formuł
- Dodanie możliwych wyników sesji doskonalenia
- Dodanie opisu roli architekta na etapie programowania
- Model skalowania zespołu architektury
- Dodano przykłady dobrych praktyk