

O'REILLY®

Helion 

Wzorce komunikacji

Przewodnik dla programistów
i architektów



Jacqui Read

Tytuł oryginału: Communication Patterns: A Guide for Developers and Architects

Tłumaczenie: Piotr Rajca

ISBN: 978-83-289-1194-9

© 2024 Helion S.A.

Authorized Polish translation of the English edition of *Communication Patterns*

ISBN 9781098140540 © 2024 Read the Architecture, Ltd.

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

Polish edition copyright © 2024 by Helion S.A.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<https://helion.pl/user/opinie/wzokop>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 230 98 63

e-mail: helion@helion.pl

WWW: <https://helion.pl> (księgarnia internetowa, katalog książek)

Printed in Poland.

- Kup książkę
- Poleć książkę
- Oceń książkę

- Księgarnia internetowa
- Lubię to! » Nasza społeczność

Spis treści

Wstęp	13
-------------	----

Część I. Komunikacja wizualna 21

1. Podstawy komunikacji	23
Poznaj swoich odbiorców	23
Mieszanie poziomów abstrakcji	27
Spójność reprezentacji	31
Podsumowanie	35
2. Uporządkuj bałagan	36
Przeciążenie kolorami	36
Ramki w ramach w ramach	38
Pajęczyna powiązań	41
Równoważenie tekstu	44
Podsumowanie	47
3. Dostępność	48
Opieranie komunikacji na kolorach	48
Dołączaj legendę	53
Odpowiednie etykiety	56
Podsumowanie	58
4. Narracja	59
Ogólny obraz jest najważniejszy	59
Dopasuj przepływ diagramu do oczekiwań	62
Jasne powiązania	66
Podsumowanie	69

5. Notacja	70
Używanie ikon do przekazywania znaczeń	70
Używanie UML dla samego UML	71
Mieszanie zachowania i struktury	75
Działanie wbrew oczekiwaniom	77
Podsumowanie	80
6. Kompozycja	81
NiezYTElne diagramy	81
Komunikacja z wykorzystaniem stylu	85
Myląca kompozycja	88
Tworzenie wizualnej równowagi	93
Podsumowanie	96

Część II. Komunikacja wielomodalna 99

7. Komunikacja pisemna	101
Prosty język	101
Piekło akronimów	103
Pisanie strukturalne	104
Składnia tekstów technicznych	108
Silne czasowniki	108
Krótkie zdania	108
Precyzyjne akapity	109
Spójne słownictwo	109
Empatia odbiorców	110
Podsumowanie	111
8. Komunikacja werbalna i niewerbalna	112
Kodowanie wiadomości	112
Korzystanie z „przepowiedni akceptacji”	112
Poświęcanie pełnej uwagi	113
Korzystanie z mowy ciała i gestów	114
Dekodowanie wiadomości	116
Walka z uprzedzeniami	116
Bycie obecnym	118
Świadomość różnic kulturowych	119
Wywieranie wpływu i perswazja	120
Podsumowanie	123

9. Trójkąt retoryczny	124
Ethos	125
Przedstaw swoje kwalifikacje	125
Korzystaj z wiarygodnych źródeł	126
Bądź transparentny	128
Zademonstruj swoją wiedzę	129
Pathos	130
Opowiedz historię	130
Mów od serca	133
Używaj żywego języka i sugestywnych obrazów	134
Logos	135
Używaj danych i faktów	135
Twórz połączenia logiczne	136
Używaj rozumowania i argumentacji	136
Podsumowanie	138

Część III. Przekazywanie wiedzy **139**

10. Zasady zarządzania wiedzą	141
Produkty ponad projektami	141
Nastawienie na projekt	142
Nastawienie na produkt	142
Formy przekazu informacji w tekście	145
Listy	146
Tabele	147
Wizualne formy przekazu informacji w tekście	148
Chmury słów	149
Wykresy, grafy i diagramy	150
Inne formy przekazu informacji w tekście	151
Dokumentacja oparta na perspektywie	151
Perspektywy bez powtórzeń	152
Perspektywy fraktalne	153
Tworzenie perspektyw	154
Podsumowanie	156
11. Wiedza i ludzie	157
Zbieraj informacje zwrotne wcześnie i często	157
Dziel się obciążeniem	160
Formaty otwarte	161
Dostępność	163
Współpraca	164

Role i obowiązki	165
Inne techniki	166
Architektura „na bieżąco”	166
Podsumowanie	170

12. Efektywne praktyki	171
ADR	171
Struktura ADR	172
Zawartość ADR	176
Przechowywanie ADR	180
Kultura ADR	181
Cechy architektury	183
Cała dokumentacja w formie kodu	187
Dokumentacja techniczna	188
Dokumentacja generowana automatycznie	190
Inne rodzaje dokumentacji	192
Podsumowanie	194

Część IV. Komunikacja zdalna **195**

13. Czas zdalny	197
Synchronizuj czas	197
Strefa czasowa	198
Empatia i kompromis	201
Dzielenie czasu pracy	201
Szanuj wzorce pracy	203
Informuj o dostępności	204
Chroń swoje godziny pracy	204
Uwzględniaj święta	205
Miej wzgląd na geografę i kulturę	206
Oszacowanie rzeczywistej wydajności pracy	207
Popraw energię i produktywność	208
Kontroluj powiadomienia	208
Automatyzuj zadania	209
Pracuj w zgodzie z rytmem innych osób	210
Planuj pod kątem poziomu energii	211
Podsumowanie	212
14. Zasady komunikacji zdalnej	213
Spotkania jako sposób synchronizacji	213
Synchroniczne kontra asynchroniczne	213
Ulepsz spotkania	215

Asynchroniczność do przemyslenia	220
Zalety komunikacji asynchronicznej	220
Przeszkody komunikacji zdalnej	220
Kierunek ma znaczenie	221
Metody komunikacji asynchronicznej	223
Rozszerzanie komunikacji asynchronicznej	225
Nastawienie na pracę zdalną	227
Przed wszystkim zdalna czy zdalna akceptowana?	227
Zalety modelu nastawionego na pracę zdalną	229
Dochodzenie do modelu nastawionego na pracę zdalną	230
Podsumowanie	233
15. Kanały komunikacji	234
Symetryczna poczta elektroniczna	234
Powody stosowania poczty elektronicznej	235
Oczekiwania dotyczące e-maili	235
Przejrzystość e-maili	236
Porady dotyczące e-maili	237
Zdalne prezentacje	239
Zaangażowanie odbiorców	239
Treść prezentacji	241
Udostępnianie ekranu	242
Narzędzia pracy zdalnej i zarządzanie nimi	242
Techniki wyboru	243
Narzędzia pracy zdalnej	244
Rozprzestrzenianie danych	247
Bezpieczeństwo	248
Wydajność narzędzi	249
Zarządzanie narzędziami	250
Podsumowanie	257
Epilog	259
A Szablon dokumentu ADR	261
Struktura ADR	261
Identyfikator i tytuł (oświadczenie o podjętej decyzji)	261
Opcje ADR	262

Wiedza i ludzie

Oprogramowanie i architektura ostatecznie sprowadzają się do ludzi. Ludzie używają oprogramowania, z kolei oprogramowanie pomaga ludziom i także ludzie projektują i piszą oprogramowanie. Nie powinno zatem dziwić, że niektóre wzorce wiedzy obracają się wokół ludzi.

Twoi współpracownicy, członkowie zespołu oraz inne osoby są zasobami nie tylko dla firmy, ale również dla Ciebie. Korzystaj z ich możliwości mądrze, a pomogą Ci usprawnić zarządzanie wiedzą, dokumentację i ogólną architekturę oprogramowania.

Zbieraj informacje zwrotne wcześniej i często

Jeden z błędów popełnianych przez wiele osób polega na tym, że poświęcają dużo czasu i wysiłku na swoją pracę, zanim otrzymają jakiegokolwiek informacji zwrotnej na jej temat. Efektem takiego podejścia może być zmarnowanie wysiłku i pieniędzy, może ono również wpłynąć niekorzystnie na projekt architektoniczny systemu. Dotyczy to zarówno osób, jak i zespołów.

Jeśli masz doświadczenie z metodami agile i przyczynami stojącymi za jej stosowaniem, prawdopodobnie wiesz, że jej założenia opierają się na wczesnym uzyskiwaniu informacji zwrotnych w najszybszy możliwy sposób. Zmiany iteracyjne i przyrostowe. Porażki szybkie lub trudne. Tych samych zasad warto przestrzegać podczas tworzenia artefaktów i dokumentacji.

Jeśli nie otrzymujesz informacji zwrotnych na temat swoich pomysłów i projektów, tracisz możliwość zmieniania wymagań oraz drugą parę oczu, która mogłaby Ci pomóc w sprawdzeniu poprawności projektu. Weź pod uwagę *efekt motyla*: jedno błędne założenie na wczesnym etapie może skierować architekturę w zupełnie złym kierunku.

Rezygnowanie z gromadzenia informacji zwrotnych można by porównać ze świadomym i celowym wpadaniem w *pułapkę utopionych kosztów*: im dłużej nad czymś pracujesz, tym mniej chcesz wprowadzać w tym zmiany. To Twoje dziecko. Jeśli nie otrzymałeś informacji zwrotnej na wczesnym etapie, jest mniej prawdopodobne, że będziesz jej szukać, i możesz przegapić kluczowy wkład w projekt, diagram itp.

Pułapka utopionych kosztów

Pułapka utopionych kosztów (ang. *sunk cost fallacy*) to błąd poznawczy, w którym osoby nadal inwestują zasoby (takie jak czas, pieniądze i wysiłek) w projekt lub decyzję pomimo dowodów na to, że inwestycja nie jest już uzasadniona lub że projekt raczej nie odniesie sukcesu. Ten błędny sposób myślenia jest zakorzeniony w przekonaniu, że ponieważ zasoby zostały już zainwestowane, dalsze inwestowanie jest konieczne, aby początkowa inwestycja nie poszła na marne. Zainwestowałeś zbyt wiele, by zrezygnować.

W projektach technicznych pułapka utopionych kosztów może przejawiać się na kilka sposobów. Przykładowo firma może kontynuować realizację projektu, który przekroczył już swój pierwotny budżet lub harmonogram, nawet jeśli dowody wskazują, że projekt nie jest już wykonalny lub nie przyniesie zamierzonej wartości. Albo zespół może kontynuować korzystanie z technologii lub narzędzia, które nie są już skuteczne ani wydajne, po prostu dlatego, że już zainwestowano w nie zasoby.

Zdobywaj informacje zwrotne na temat małych części i całości projektu architektury. Niekiedy tworzenie diagramu lub zestawu diagramów dla części architektury może zająć dużo czasu, nawet kilka dni. Wyobraź sobie, iż wkładasz trzy dni wysiłku w diagramy tylko po to, aby się dowiedzieć, że przyjęte założenie było błędne lub że Twoja opinia była fałszywa. W najgorszym przypadku konieczne może być rozpoczęcie pracy od zera, a nawet nakazanie zespołowi programistów zaprzestania pracy opartej na tych diagramach. Nawet w najlepszym przypadku będziesz musiał poświęcić więcej czasu i wysiłku na zmianę diagramów.

Rozważ następujące dodatkowe korzyści, jakie przynosi osobom i zespołom otrzymywanie informacji zwrotnych wcześniej i często:

Wczesna identyfikacja problemów i błędów

Im więcej czasu upływa od podjęcia decyzji lub działania do momentu ich zakwestionowania, tym wyższy jest koszt wprowadzenia zmiany. Na pewno zależy Ci, by identyfikować problemy, zanim ich naprawa stanie się bardziej kosztowna¹. Rysunek 11.1 ilustruje sytuację, której można uniknąć, uzyskując informacje zwrotne.

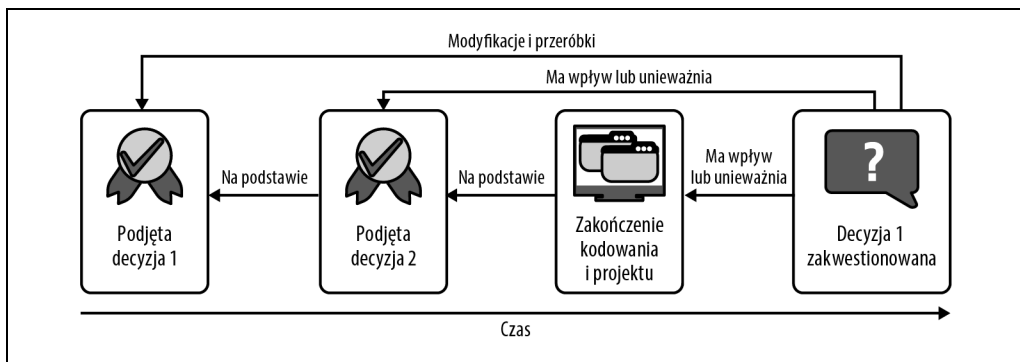
Identyfikacja możliwych ulepszeń i optymalizacji

Zaangażowanie większej liczby osób o różnym pochodzeniu i doświadczeniu oznacza bardziej holistyczną i zróżnicowaną analizę projektów, dzięki czemu jest bardziej prawdopodobne, że dostrzeżesz miejsce na ulepszenia dzięki informacjom zwrotnym, niż wówczas, gdybyś tymi informacjami nie dysponował.

Zapewnienie zgodności z potrzebami biznesowymi

Interesariusze biznesowi lub inne osoby bliżej związane z tymi zagadnieniami mogą dostarczyć Ci informacji zwrotnych na temat potrzeb biznesowych, które ma spełniać Twój projekt.

¹ Stosowanie ADR to dobra technika pomagająca zidentyfikować problemy przed podjęciem ważnej decyzji.



Rysunek 11.1. Brak gromadzenia informacji w momencie podejmowania decyzji 1. prowadzi do późniejszych kosztownych zmian

Nawiązanie dialogu

Interesariusze docenią zaangażowanie i zwrócenie uwagi na kompromisy i negatywy wynikające z ich wymagań. Mogą nawet zmienić swoje wymagania w oparciu o informacje zwrotne, co powinno nastąpić raczej wcześniej niż później.

Identyfikacja ryzyka i wyzwań oraz podejmowanie odpowiednich działań

Im wcześniej ryzyko zostanie zidentyfikowane, tym lepiej. Zagrożenia i wyzwania można proaktywnie ograniczać, a także uzyskać dalsze informacje zwrotne w celu zaakceptowania ryzyka, którego nie można było ograniczyć.



Dobrym pomysłem jest uzyskanie informacji zwrotnej od osób, które nie są zaangażowane w projekt. Mogą one dać zewnętrzną perspektywę, która pomoże uniknąć kłótwy wiedzy i zburzyć ściany komory echa.

Kiedy więc warto otrzymywać informacje zwrotne? Ogólna odpowiedź brzmi: wcześniej i często, ale kiedy dokładnie, będzie to zależec od sytuacji. Jednym z kluczowych momentów na uzyskanie informacji zwrotnej jest dokumentowanie założeń (aby zagwarantować, że wszystkie zostaną udokumentowane).

Założenia należy jak najszybciej zatwierdzić (potwierdzić), zanim wpłyną na projekt systemu lub produktu. Nie zawsze jest to możliwe, więc jeśli nie możesz uzyskać podpisu na założeniu, wyraźnie zaznacz to w dokumentacji. Nadaj swoim założeniom identyfikatory w dokumentacji, aby skutecznie się do nich odwoływać, dzięki czemu będziesz wiedzieć, gdzie mogą być potrzebne zmiany, jeśli założenie okaże się fałszywe.

Jeśli od niedawna zajmujesz się pracą nad danym produktem lub na danym stanowisku, radzę jak najczęściej uzyskiwać opinie od kolegów z działów technicznego i biznesowego. W perspektywie dłuższego czasu pozwoli Ci to zaoszczędzić czas i poprawić reputację. Frustracja spowodowana prośbami o opinie będzie krótkotrwała, podczas gdy koszty wizerunkowe związane z wyprodukowaniem czegoś, co ostatecznie jest nieodpowiednie lub kosztowne, na pewno będą trwalsze.

Ale nie myśl, że tylko dlatego, iż masz już za sobą lata pracy albo piastujesz kierownicze stanowisko, nie musisz szukać informacji zwrotnych u swoich współpracowników. Nawet najmłodszy członek zespołu może wiedzieć coś, czego Ty nie wiesz, lub mieć inną perspektywę niż Ty. Wysokie stanowisko nie może stanowić przeszkody na drodze do zdobywania mądrości.



Analizowanie założeń i potrzeb interesariuszy buduje zaufanie do projektów i rozwiązań, pokazując, że ich słuchasz. Rzadko kiedy projekt staje się rzeczywistością, jeśli brakuje do niego zaufania.

Teraz, gdy już wiesz, dlaczego i kiedy uzyskać informacje zwrotne, zapewne chcesz również dowiedzieć się, *jak* to zrobić. Pamiętaj, że o informacje zwrotne *nie musisz* wyraźnie prosić. Możesz po prostu pokazać ludziom, nad czym pracowałeś, aby umożliwić im wyrażenie opinii, oczywiście jeśli będą chcieli to zrobić. Należy pamiętać, że osoby młodsze od Ciebie mogą potrzebować zachęty do wyrażenia opinii, więc najlepiej jest wprost o nią poprosić.

Dostępnych jest wiele metod uzyskiwania informacji zwrotnych. Oto kilka przykładów:

- Najłatwiejszym podejściem jest po prostu zapytać lub pokazać swoją pracę bez wywierania presji na odbiorcach, by udzielili informacji zwrotnej.
- Informacje zwrotne można włączyć do formalnych procesów, takich jak prośba o zaakceptowanie zmian (ang. *pull request*), lub dodać do punktów kontrolnych w osobistym lub zespołowym przepływie pracy.
- Dokumenty ADR są doskonałym narzędziem do uzyskiwania informacji zwrotnych na temat proponowanych decyzji². Włączenie sekcji opinii lub konsultacji, w której inni mogą dzielić się poradami, oraz wyznaczenie terminu na wniesienie wkładu umożliwi innym zapoznanie się z decyzją i jej uzasadnieniem oraz wyrażenie własnej opinii na ich temat.
- Wszelkie spotkania, stand-upy i przeglądy są dobrą okazją do poruszenia kwestii, nad którymi pracujesz, w celu uzyskania nieformalnej lub formalnej opinii.
- Jeśli w firmie istnieje proces wnioskowania o zmianę (ang. *request for change*, RFC), może on stanowić oczekiwaną metodę uzyskiwania opinii na temat pomysłów na zmiany w produkcie lub systemie.

Dziel się obciążeniem

Tworzenie i utrzymywanie dokumentacji (i ogólna komunikacja samej architektury oprogramowania) nie powinno należeć tylko do jednej osoby (chyba że faktycznie sam tworzysz jednoosobowy zespół). Żadna rola nie jest obciążona wyłączną odpowiedzialnością za te działania — każda z nich odpowiada zwykle za przygotowanie różnych typów lub elementów dokumentacji. Zarówno tworzenie, jak i utrzymywanie dokumentacji musi być współdzielone, aby była ona skuteczna i aktualna. Przyjrzyjmy się, jak można podzielić się tym obciążeniem.

² Więcej informacji na ten temat można znaleźć w podrozdziale „ADR” w rozdziale 12.

Zbieranie informacji zwrotnych jest elementem procesu

W zespołach programistycznych, architektonicznych i technicznych w Polyglot Media gromadzenie informacji zwrotnych stanowi element przepływu pracy. Zadania projektowe i dokumentacyjne są podzielone na małe i logiczne części, w taki sam sposób jak zadania związane z pisaniem kodu lub opracowywaniem historyjek użytkowników. W rezultacie cykl publikacji i gromadzenia informacji zwrotnych dla artefaktów i dokumentacji jest szybki i zwinny.

Każdy członek zespołu może poprosić o recenzję za pośrednictwem specjalnego kanału w swoim systemie wiadomości, przy czym oczekuje się, że informacja zwrotna zostanie odesłana w krótkim czasie. Wszyscy są zachęceni do stosowania tego sposobu pozyskiwania informacji zwrotnych, zanim zadanie zostanie uznane za zakończone.

Formalne informacje zwrotne są wymagane w następujących punktach kontrolnych:

- Gdy składane jest żądanie pobrania dokumentacji do repozytorium lub gdy artefakt jest uznawany za gotowy, opinie są pozyskiwane od współpracowników, a w razie potrzeby od programistów i architektów są pozyskiwane opinie techniczne.
- Po przygotowaniu *zapoznawczej* wersji dokumentacji (na przykład gdy w środowisku innym niż produkcyjne zostanie opublikowana strona internetowa z dokumentacją) pozyskiwane są opinie interesariuszy, takich jak właściciel produktu.
- Po opublikowaniu dokumentacji i artefaktów gromadzone są opinie od klientów, zespołu wsparcia i innych współpracowników.

W Polyglot Media gromadzenie informacji zwrotnych jest uważane za ciągły i zwinny proces.

Formaty otwarte

Korzystanie z ogólnodostępnych aplikacji i otwartych formatów plików to świetny sposób na rozszerzenie liczby osób, które mogą tworzyć i aktualizować (a także przeglądać) dokumentację. Korzystanie z otwartych formatów i ogólnodostępnych aplikacji skutkuje następującymi korzyściami:

Mniej problemów z licencją

Nie musisz martwić się o licencje (przynajmniej aż tak bardzo), na przykład: kto w zespole potrzebuje pełnej licencji Visio lub kto potrzebuje konta Atlassian oraz jakimi uprawnieniami osoby te muszą dysponować (obie te licencje są bardzo drogie i generują dalsze koszty związane z zarządzaniem licencjami). Należy jednak sprawdzić licencję każdego używanego produktu.

Większa dostępność edytorów

Łatwiej jest uzyskać pozwolenie na korzystanie z jednego z wielu możliwych edytorów dla każdego, kto musi tworzyć lub edytować dokumentację. Wiele niezastrzeżonych aplikacji jest darmowych, nawet komercyjnie. Często istnieje kilka opcji aplikacji, które mogą odczytywać i zapisywać niezastrzeżony format pliku. Niektóre aplikacje działają nawet w przeglądarce (np. draw.io), więc nie muszą być instalowane przez użytkownika.

Większa interoperacyjność

Ponieważ format pliku nie jest własnością firmy lub osoby, jest otwarty i publicznie udokumentowany, każdy może opracować aplikację, która będzie w stanie odczytywać i zapisywać pliki w danym formacie. A to z kolei oznacza, że możesz używać swoich artefaktów w więcej niż jednej aplikacji.

Ograniczona blokada dostawcy

Posiadanie całej dokumentacji w formacie Markdown (lub innym prostym formacie tekstowym)³ oznacza, że każdy, kto ma edytor tekstu, może ją tworzyć, czytać i aktualizować. Każdy, kto tego potrzebuje, będzie dysponował wieloma aplikacjami pozwalającymi odczytywać i zapisywać pliki w niezastrzeżonym formacie. Nie jest to prawdą w przypadku dokumentacji tworzonej w programie Word bądź Notion lub zapisywanej w innym zastrzeżonym formacie.

Nie musisz i prawdopodobnie nie możesz unikać oprogramowania własnościowego, ale powinno ono być brane pod uwagę w przypadku wszystkiego, co trwałe (na przykład dokumentacji, która będzie potrzebna w przyszłości). Oprogramowanie ogólnodostępne jest prawdopodobnie bardziej przyszłościowe niż oprogramowanie własnościowe. Jak w przypadku wszystkiego, istnieją kompromisy, takie jak brak wsparcia dla wielu aplikacji otwartych i programów open source. Przy wyborze narzędzi należy rozważyć wszystkie kompromisy.

Formaty własnościowe

Oprogramowanie ogólnodostępne i otwarte formaty plików nie są własnością ani nie są kontrolowane przez konkretną firmę lub osobę. Takie oprogramowanie jest zazwyczaj rozwijane i dystrybuowane na zasadach licencji open source.

Otwarte formaty plików są otwarte i publicznie udokumentowane, umożliwiając wszystkim tworzenie oprogramowania, które może dany format odczytywać i zapisywać. Oto przykłady ogólnodostępnego oprogramowania i otwartych formatów plików:

Markdown

Prosty format tekstowy obsługiwany przez wiele edytorów i narzędzi do kompilacji.

AsciiDoc

Format tekstowy umożliwiający przekształcanie zapisanych w nim plików w sformatowane dokumenty przy użyciu takich narzędzi jak AsciiDoctor i Antora.

Git

Darmowy i otwarty rozproszony system kontroli wersji.

ODF (Open Document Format)

Otwarty format dokumentów edytora tekstu, arkuszy kalkulacyjnych, prezentacji i rysunków graficznych.

³ Więcej informacji o formacie Markdown można znaleźć w podrozdziale „Cała dokumentacja jako kod” w rozdziale 12.

draw.io

Otwarta aplikacja i format plików graficznych.

PNG (Portable Network Graphics)

Otwarty format grafiki rastrowej będący alternatywą dla formatu GIF (Graphics Interchange Format).

PDF (Portable Document Format)

Format plików do prezentacji tekstu i grafiki niezależny od oprogramowania i sprzętu.

YAML (YAML Ain't Markup Language)

Otwarty, czytelny dla człowieka język serializacji danych, często używany do zapisu metadanych i plików konfiguracyjnych.

HTML (Hypertext Markup Language)

Standardowy język znaczników do tworzenia dokumentów przeznaczonych do przeglądania w przeglądarce internetowej.

Dostępność

Wybierając format lub notację dla dokumentacji, należy wziąć pod uwagę, czy może on być używany przez wszystkie osoby, które muszą czytać, tworzyć i aktualizować artefakty. Oprócz wyboru otwartego formatu należy wziąć pod uwagę zarówno odbiorców, jak i wiedzę i możliwości autora.

Standard taki jak UML lub ArchiMate może ostatecznie ograniczyć dostęp. Liczba osób dysponujących pełną znajomością lub kwalifikacjami w zakresie tych standardów jest niewielka⁴. Konieczne byłoby zainwestowanie czasu i pieniędzy w szkolenia, aby zwiększyć grono osób mogących tworzyć dokumentację, i powtarzanie tego za każdym razem, gdy ktoś nowy dołączy do zespołu (zobacz rysunek 11.2).

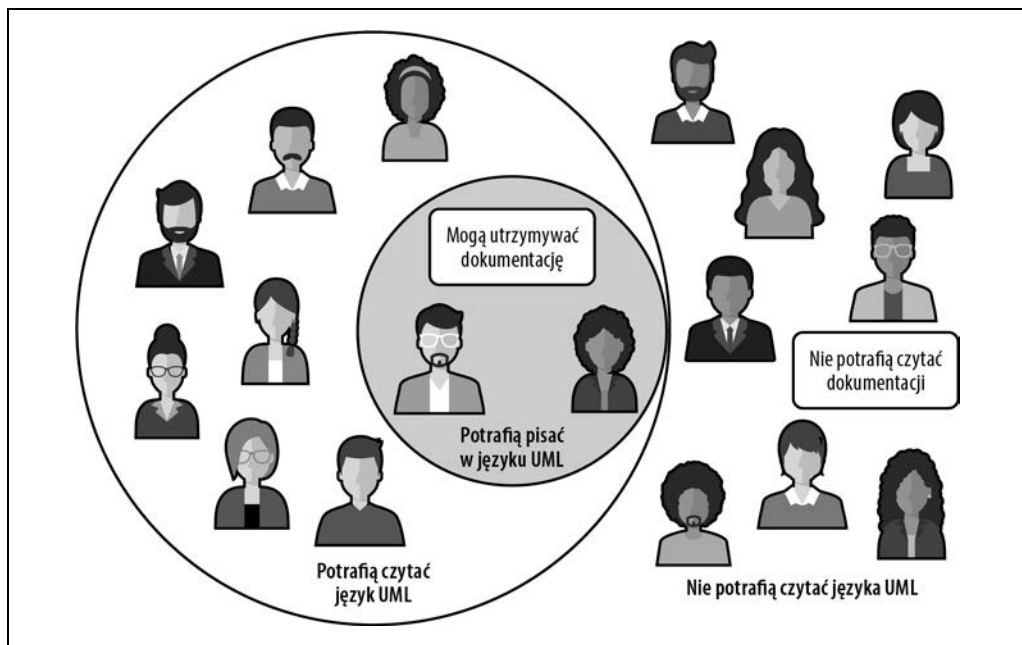


Płacenie za własnościowe narzędzia, takie jak Microsoft Word i Google Docs, łatwo może stać się przykrą, lecz niezauważalną koniecznością, jak tankowanie pojazdu. Przejście na niezastrzeżony format, który jest powszechnie obsługiwany, taki jak Markdown, oznacza, że możesz przestać płacić i zacząć korzystać z niezliczonych bezpłatnych narzędzi. Oszczędzasz pieniądze i zyskujesz wiele innych korzyści, takich jak lepsza zgodność⁵.

Prosta niestandardowa notacja, którą opracujesz we własnym zakresie, będzie łatwiejsza do przyswojenia dla każdego. Niektóre standardy, takie jak C4 Model, są również znacznie łatwiejsze do zrozumienia bez wcześniejszej wiedzy, można się ich łatwo nauczyć i wykorzystać je do tworzenia

⁴ Ten sam problem występuje zazwyczaj w przypadku wykorzystywania własnościowego narzędzia lub standardu, nie dotyczy tylko tworzenia diagramów.

⁵ Jeśli kiedykolwiek znalazłeś się w sytuacji, w której jedna osoba korzystała z programu Microsoft Word, a druga używała aplikacji takiej jak LibreOffice Writer do edycji tego samego dokumentu, zrozumiesz, że format *docx* programu Word nie ma wysokiej zgodności.



Rysunek 11.2. Wybór notacji lub formatu ma wpływ na to, kto może zrozumieć lub prowadzić dokumentację

i aktualizowania dokumentacji. Korzystanie z prostych standardów, takich jak diagramy przepływu lub uproszczone diagramy sekwencji, zwiększy grono osób, które będą mogły zajmować się dokumentacją.

Ten wybór formatów i standardów dotyczy całej dokumentacji. Wyobraź sobie, że każdy, kto ją tworzy i utrzymuje, musiałby być w stanie pisać w języku HTML i używać kaskadowych arkuszy stylów (CSS). Byłoby to trudne zadanie, wymagające dużej znajomości składni i logiki. A teraz porównaj to z tworzeniem dokumentacji w formacie Markdown lub AsciiDoc. Oba te formaty pozwalają na dużą kontrolę nad wyglądem tekstu na ekranie, ale są znacznie prostsze do zrozumienia i praktycznego opanowania.



Poczta elektroniczna nie jest repozytorium wiedzy. Wiadomości e-mail są dostępne tylko dla nadawcy i odbiorcy. Trudno je przeszukiwać. Wiadomości są zwykle usuwane, gdy ktoś opuszcza organizację, a wiele organizacji regularnie usuwa stare wiadomości. Dlatego przenoś wiedzę z poczty elektronicznej do lokalizacji, w której będzie bezpieczna, możliwa do przeszukiwania i dostępna dla wszystkich, którzy jej potrzebują.

Współpraca

Innym sposobem na usprawnienie prowadzenia dokumentacji jest korzystanie z narzędzi do współpracy. Dokumenty Google, Microsoft Teams, Slack i tablice online umożliwiają łatwe współdziałanie w tworzeniu i utrzymywaniu artefaktów i dokumentacji. Współpraca przy użyciu takich narzędzi może odbywać się synchronicznie (w tym samym czasie) lub asynchronicznie (w różnym czasie). Obie metody są przydatne w różnych okolicznościach.

Pracując wspólnie nad tworzeniem lub utrzymywaniem artefaktu, można zmniejszyć obciążenie, uzyskując dane wejściowe od więcej niż jednej osoby naraz. Można nawet zmieniać osobę *kierującą* (mającą kontrolę nad klawiaturą, myszą lub innym urządzeniem wejściowym), tak samo jak robią programiści podczas programowania w parach lub zespołach. Jeśli nikt nie chce wykonać określonego zadania, zróbcie to razem.

Wspólna praca oznacza również, że można się wzajemnie szkolić lub uczyć, co może być szczególnie przydatne dla nowych członków zespołu. Jeśli chcesz delegować zadanie, które tylko Ty obecnie rozumiesz, skorzystanie z narzędzia do współpracy jest doskonałą opcją — zwłaszcza jeśli wszyscy uczestnicy pracują zdalnie.

Narzędzia do współpracy, takie jak Slack i Microsoft Teams, są również doskonałym sposobem na powiadamianie lub informowanie innych. Przydatne może być skonfigurowanie automatycznego powiadamiania w ramach kanału lub zespołu, rozsyłającego informacje, gdy dokumentacja zostanie zaktualizowana lub utworzona⁶. W najprostszym przypadku pomoże to wykształcić nawyki związane z utrzymaniem dokumentacji. Być może zechcesz także skorzystać z możliwości automatycznego powiadamiania o zmianach (czy to w kodzie, na przykład gdy zostanie udostępniona nowa wersja, czy to w dokumentacji, na przykład gdy zostaną zmienione opisane w niej wymagania). Takie rozwiązanie może zmuszać zespół do sprawdzania dalszych skutków zmiany (na przykład czy zmiana wymagań wprowadzona przez analityka biznesowego będzie wymagać wprowadzenia zmian w architekturze przez architekta lub programistę).

Role i obowiązki

Przypisanie konkretnych ról związanych z dokumentacją i komunikacją każdej osobie w zespole może być przydatne, ale nie należy wpadać w pułapkę tworzenia wąskich gardeł (na przykład kilka osób odpowiedzialnych za większość pracy) lub pojedynczych punktów awarii (na przykład członek zespołu jest chory lub odchodzi z firmy). Upewnij się, że masz przynajmniej dublera dla każdej roli, aby ktoś mógł przejąć obowiązki lub pomóc w razie potrzeby. Rozważ obciążenie pracą każdej konkretnej roli i przypisz do niej odpowiednią liczbę osób.

Jednym ze sposobów tworzenia ról związanych z dokumentacją jest powiązanie ich z typami dokumentacji, takimi jak rejestrowanie wymagań lub wyników sesji EventStorming. Innym sposobem jest przypisanie roli odpowiedzialnej za tworzenie dokumentacji konkretnego typu jednej osobie i roli odpowiedzialnej za przeglądanie tego typu dokumentacji komuś innemu. Niektóre rodzaje dokumentacji będą naturalnie dopasowane do ról w zespole, a inne mogą bardziej pasować osobom, które dysponują odpowiednimi umiejętnościami lub czasem.



Nigdy nie zakładaj, że będziesz mieć okres przejściowy na przekazanie obowiązków innemu członkowi zespołu. Upewnij się, że nie występuje żaden pojedynczy punkt awarii (tylko jedna osoba, która za coś odpowiada), jeśli chodzi o dokumentację i komunikację. Nigdy nie wiadomo, kiedy ktoś może pójść na długotrwałe zwolnienie lekarskie lub urlop albo zostać zwolniony.

⁶ To powiadamianie może być ręczne, ale zapewne będziesz chciał zautomatyzować wszystko, co możesz.

Inne techniki

Dzielenie się wiedzą lub lunch połączony z nauką to świetne sposoby na rozpowszechnianie wiedzy w całym zespole lub organizacji, przy czym powinny one opierać się na dokumentacji lub innych zasobach, w których można znaleźć te same informacje (bądź też przygotowanie takich sesji powinno obejmować tworzenie dokumentacji lub zasobów). Takie sesje mogą być prowadzone na żywo lub nagrywane, ale jeśli odbywają się na żywo, warto podczas nich tworzyć artefakty, z których będą mogli uczyć się przyszli członkowie zespołu lub pracownicy. Wysiłku włożonego w przygotowanie jednej sesji nie powinno się marnować poprzez uczynienie z niej jednorazowej okazji do nauki. Sesje takie nie tylko pozwalają dzielić się wiedzą, ale także przyczyniają się do wykształcania kultury dzielenia się i nauki.

W przypadku artefaktów tworzonych wielokrotnie dla jednego produktu lub projektu albo dla wielu projektów i produktów sensowne jest użycie szablonu. Szablony są podobne do wzorców: każdy szablon jest rozwiązaniem wielokrotnego użytku, które okazało się skuteczne.

Korzystanie z szablonów ma wiele zalet. Jedną z nich polega na tym, że ułatwiają one współdzielenie obciążenia. Gdy dostępny jest szablon, to nawet jeśli w zespole pojawia się nowa osoba, ma ona od czego zacząć, a kiedy szablony są współdzielone przez produkty lub projekty, to dzięki wcześniejszym doświadczeniom w ich stosowaniu każdy, kto zmieni produkt lub projekt, będzie mógł szybko wdrożyć się do pracy.

Niezależnie od tego, czy pracujesz w systemie agile, czy nie, sprint dokumentacyjny lub po prostu specjalny blok czasowy to kolejny sposób na dzielenie się pracą nad dokumentacją i komunikacją. Ten czas oznacza, że osoby, które normalnie mogą być zbyt zajęte, nie mają wymówki, aby wymigać się od tworzenia lub utrzymywania dokumentacji. Z takim blokiem czasowym można połączyć techniki takie jak dzielenie się wiedzą lub sesje współpracy, aby uzyskać dodatkowy efekt, zwłaszcza jeśli sesje te zwykle nie odbywają się poza tą przestrzenią.



Jeśli chcesz pozwolić członkom swojego zespołu na współdzielenie i delegowanie komunikacji i dokumentowania architektury oprogramowania i produktu, musisz mieć proaktywne podejście do dokumentacji. Podziel się obciążeniem i spraw, aby tworzenie dokumentacji i komunikacja były wspólnym wysiłkiem.

Architektura „na bieżąco”

Nie będziesz tego potrzebować (ang. *you aren't gonna need it*, YAGNI) to zasada programistyczna, która zachęca do rozwijania tylko tych funkcji, które są potrzebne w danej chwili, i odradza podejmowanie prób przewidywania przyszłych potrzeb⁷. Jest to kolejna zasada związana z pisaniem kodu, która powinna być stosowana także do zarządzania wiedzą i dokumentacją.

⁷ Nie stanowi ona jednak wymówki, by nie stosować najlepszych praktyk.



Zasada YAGNI opiera się na założeniu, że prognozy mogą się nie sprawdzić, a zatem robiąc coś, co nie jest aktualnie potrzebne, marnujesz wysiłek i możesz doprowadzić do problemów w przyszłości, gdy trzeba będzie wprowadzić zmiany.

Inną zasadą (tym razem dotyczącą architektury oprogramowania), która również dobrze pasuje do zasady YAGNI, jest odraczanie decyzji architektonicznych tak długo, jak to możliwe. Łącząc obie te zasady, otrzymujemy coś, co można by określić jako architekturę i dokumentację wykonywaną „na bieżąco” (wtedy, gdy jest konieczna — ang. *just in time*): nie decyduj i nie dokumentuj tego, co wydaje Ci się potrzebne w przyszłości, lecz jedynie to, co wiesz, że jest potrzebne już *teraz*.

Postępowanie zgodnie z tym wzorcem ma następujące zalety, i to niezależnie od tego, czy pracujesz zgodnie z metodą kaskady, agile czy jakimś rozwiązaniem pośrednim:

Redukcja marnotrawstwa

Kiedy tworzysz architekturę lub dokumentację na bieżąco, opracowujesz ją z jakiegoś powodu i z całą wiedzą dostępną w danym momencie. Jeśli zaś tworzysz ją z wyprzedzeniem, prawdopodobnie zmieni się coś, co będzie wymagało jej zmiany. W przypadku gdy jakaś inna decyzja, artefakt, projekt lub kod bazuje na tym zmienionym elemencie, pojawi się łańcuch elementów, które także będą musiały ulec zmianie. A to oznacza wiele zmarnowanego wysiłku, który na pewno będzie kosztowny.

Większa zwinność i elastyczność

Wymagania zmieniają się, a dzięki zastosowaniu wzorca architektury „na bieżąco” można w dużym stopniu reagować na te zmiany. Podczas wykonywania pracy masz najnowsze informacje na temat wymagań i decyzji. Ponadto stosowanie systemu „na bieżąco” daje Ci dobrą pozycję do reagowania na zmiany, które mogą wymagać aktualizacji poprzednich artefaktów, gdyż proces planowania opiera się na potrzebach, a nie na przewidywaniach. Zmniejszasz liczbę artefaktów, które będą musiały zostać zmienione, a zatem masz większe szanse na to, że będziesz dysponować zasobami i możliwościami adaptacji.

Efektywne wykorzystanie zasobów

Koncentrując się na tym, co jest potrzebne teraz, a nie jutro, ustalasz priorytety działań swojego zespołu. Wszyscy członkowie zespołu, zarówno techniczni, jak i biznesowi, koncentrują się na bieżącym, najważniejszym aspekcie produktu lub projektu. Wszyscy marnują mniej czasu i wysiłku, gdy artefakty są tak aktualne, jak to tylko możliwe. A ponieważ czas wszystkich jest wykorzystywany bardziej efektywnie, istnieje większa elastyczność w zakresie badań, weryfikacji koncepcji i nieoczekiwanych prac.

Aktualne informacje

Nie tylko będziesz mieć najbardziej aktualne informacje i opinie zwrotne niezbędne do podjęcia decyzji lub stworzenia artefaktu, ale także artefakt będzie używany zaraz po jego utworzeniu, a zatem w momencie użycia będzie aktualny. Jeśli odłożysz utworzenie artefaktu lub podjęcie decyzji w czasie, będziesz wiedzieć więcej niż wcześniej. Ogólnie rzecz biorąc, dzięki zastosowaniu procesów zgodnych z podejściem „na bieżąco” zyskasz doskonałe możliwości do utrzymywania wszelkich artefaktów.

Krótszy czas wprowadzania produktu na rynek

Podejście „na bieżąco” oznacza, że cały wysiłek jest wkładany w to, co jest najważniejsze teraz, a nie w to, co może być ważne w przyszłości. Dlatego zmiany lub funkcje, które są ważne teraz, mogą być testowane i wdrażane szybciej, niż gdyby skupiono się na innych zmianach, które aktualnie nie mają znaczenia.

Lepsze dopasowanie do praktyk agile

Chociaż architektura „na bieżąco” może być stosowana w środowisku kaskadowym, to przekształca ona procesy architektoniczne w coś znacznie bardziej zwinnego. Pozwala to osobom zaangażowanym w opracowywanie architektury i dokumentacji na dopasowanie się do procesów agile zespołu programistów i poprawia ogólny cykl gromadzenia informacji zwrotnych agile, ponieważ architektura lepiej reaguje na zmiany. Architektura ta umożliwia nawet dodawanie zadań architektonicznych do tablic metod Kanban lub Scrum.

Większa przejrzystość

Utrata ogólnego obrazu wskutek zbytniego skoncentrowania się na szczegółach to przypadłość, która może występować w wielu zestawach dokumentacji. Gdy w dokumentacji znajduje się tylko to, co jest ważne teraz (lub wcześniej), użytkownik ma mniej do przeszkaniania, aby znaleźć to, czego potrzebuje.



Jeśli dokumentacja jest niepoprawna lub jej utrzymanie kosztuje znacznie więcej, niż zapewnia korzyści, ogólna wartość netto dokumentacji będzie ujemna.

Jeśli pracujesz w środowisku kaskadowym lub środowisku, które nie jest przystosowane do regularnych zmian przyrostowych, możliwe i korzystne jest zastosowanie architektury „na bieżąco”. Być może sytuacja będzie wymagać, byś zaprojektował cały system przed napisaniem jakiegokolwiek kodu, ale wciąż będziesz mógł ustalać priorytety i porządkować decyzje, które należy podjąć, oraz artefakty, które należy utworzyć. W celu uzyskania bardziej aktualnych informacji prace mogą być przeplatane badaniami lub działaniami sprawdzającymi poprawność koncepcji. Odraczaj jak najwięcej, a może się okazać, że wszystkie procesy staną się bardziej zwinne.



Możesz połączyć architekturę „na bieżąco” (ang. *just in time*) z architekturą „nie dłużej niż to konieczne” (ang. *just long enough*): wycofaj dokumentację i artefakty, które spełniły swoje zadanie, i wyraźnie zaznacz, że nie są już one aktualne. Architektura „nie dłużej niż to konieczne” oszczędza czas, jaki poświęciłbyś na utrzymywanie treści, które nie są już przydatne, i eliminuje ryzyko, że użytkownicy będą korzystali z nieaktualnych informacji.

Oczywiście zawsze istnieją kompromisy. Oto niektóre spośród tych, które należy wziąć pod uwagę podczas tworzenia architektury „na bieżąco”:

Ludzie, którzy chcą informacji teraz

Zawsze znajdzie się ktoś, kto chce uzyskać wszystkie informacje jak najszybciej. Są to osoby, które trzeba przekonać poprzez *działanie*. Rozmawiaj z nimi, lecz jednocześnie pokaż im korzyści płynące z jak najszybszego stosowania architektury „na bieżąco”. Jeśli nalegają, możesz dać

im artefakt, który jest bardzo wyraźnie oznaczony jako wersja robocza lub oczekująca, i pozwolić im zdecydować, czy chcą oprzeć własną pracę na takich wersjach roboczych lub niepewnych informacjach.

Ludzie, którzy chcą, abyś przewidywał przyszłość

Zapewne cały czas ktoś prosi Cię o prognozowanie. Jak długo to potrwa? Czy do tego momentu zdobędziemy potrzebne informacje? Rally, dostawca oprogramowania agile, wykazał, że zespoły, które szacują najmniej, produkują najwięcej⁸. Kierownik projektu musi stworzyć swój budżet i raportować swoje kamienie milowe, ale to wszystko to tylko domysły. Architektura „na bieżąco” może pomóc z takimi zagadnieniami jak długofalowe zarządzanie zasobami, ponieważ ogólnie rzecz biorąc, dzięki niej wszyscy będą bardziej elastyczni pod względem możliwości realizacji nieoczekiwanej pracy. Należy dążyć do rozbicia kamieni milowych na decyzje i zadania, aby tworzyć artefakty, które pomogą określić, czy wszystko jest na dobrej drodze.

Utrata ważnych informacji, ponieważ nie są one teraz istotne

Jeśli koncentrujesz się na podejmowaniu decyzji i tworzeniu artefaktów, które są potrzebne teraz, możesz łatwo utracić ważne pomysły i informacje, które mogą być istotne później. Utwórz miejsce do zapisywania takich informacji (np. strona na wiki). Takie informacje należy sprawdzać podczas pracy nad kolejnymi artefaktami i decyzjami.

Powody odwlekania decyzji

Maksymalne odkładanie decyzji dotyczących architektury jest dobrą praktyką z kilku powodów:

Większa elastyczność

Poprzez odłożenie decyzji na później można łatwiej dostosować się do zmieniających się wymagań i okoliczności.

Lepsza nauka

Opóźnienie decyzji oznacza więcej czasu na naukę i eksperymentowanie.

Niższe ryzyko

Zmniejszasz ryzyko podejmowania błędnych lub nieoptymalnych decyzji w oparciu o niekompletne lub niedokładne informacje.

Lepsza współpraca

Mając więcej czasu na zebranie informacji i porad, masz większe szanse na uzyskanie wkładu od różnych osób i od osób mających kluczowe znaczenie dla podejmowanej decyzji.

⁸ Zdaniem Jeffa Sutherlanda, wynalazcy i współtwórcy Scruma, „szacowanie zadań spowolni cię. Nie rób tego. Zrezygnowaliśmy z tego ponad 10 lat temu. Dziś mamy dobre dane z Rally na temat 60 000 zespołów. Najwolniejsi szacują zadania w godzinach. Brak szacowania w ogóle poprawi wydajność zespołu w porównaniu ze stosowaniem szacowania godzinowego” (<https://www.quora.com/What-are-the-techniques-set-by-the-Scrum-guidelines-for-a-task-estimation-in-sprint-planning-Are-there-any-limitations-to-these-techniques>).

Mniejsza złożoność

Odroczenie decyzji zasadniczo przekłada się na mniejszą złożoność, ponieważ skupiasz się tylko na tym, co jest aktualnie potrzebne, i nie musisz poprawiać rzeczy, które trzeba zmienić po podjęciu decyzji. Refaktoryzacja w celu dodania czegoś jest łatwiejsza niż poprawianie w celu usunięcia lub wprowadzenia zmiany.

Wyższa wydajność

Odroczenie decyzji prowadzi do usprawnienia procesu i zmniejszenia marnotrawstwa wysiłku. Gdy decyzja jest podejmowana wcześniej niż to konieczne, jest bardziej prawdopodobne, że zmiany na nią wpłyną, a to będzie oznaczać konieczność aktualizacji kodów lub projektów, które na tej decyzji bazują. Być może uda Ci się całkowicie uniknąć konieczności podejmowania niektórych decyzji po zmianie jakichś wymagań lub informacji.

Podsumowanie

Jeśli chodzi o zarządzanie wiedzą i dokumentacją, Twój koledzy i współpracownicy są niezbędnymi zasobami, a techniki i wzorce, które omówiłam w tym rozdziale, pozwolą Ci z tych zasobów korzystać.

Teraz, kiedy w swoim zestawie narzędzi posiadasz już zasady wyższego poziomu i możesz korzystać ze wsparcia innych osób, nadszedł czas, aby przyjrzeć się niektórym powszechnym praktykom zarządzania wiedzą i dokumentacją oraz temu, jak można zapewnić ich *skuteczność*.

PROGRAM PARTNERSKI

— GRUPY HELION —



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 

Oto mistrzowski przewodnik, który odblokowuje złożoność kryjącą się za językiem ludzkich interakcji!

Mark Richards, architekt oprogramowania

Skuteczna komunikacja jest równocześnie sztuką i nauką. Koszt niewłaściwej komunikacji bywa wysoki, dlatego warto zgłębiać tę umiejętność. Architekci wiedzą, jak ważne w ich fachu są *wzorce* i *antywzorce*, które można zastosować (lub rozpoznać) podczas projektowania systemów. Okazuje się, że te same koncepcje doskonale się sprawdzają w odniesieniu do komunikacji.

Ta książka jest wyjątkowo praktycznym przewodnikiem, dzięki któremu przyswoisz ważną umiejętność skutecznego komunikowania wizji, aby uzyskiwać wsparcie dla swojego projektu, a także wkład ze strony zespołów. Dowiesz się, jak przekonująco przedstawiać zamierzenia i jak unikać nieporozumień, które mogą prowadzić do wzrostu kosztów, niespełnionych wymagań czy innych niż zakładane efektów pracy. Poprzez konstruktywne przykłady i wzorce, poradnik uczy umiejętności niezbędnych do jasnego wyrażania swojego przekazu wobec różnych odbiorców, z którymi ma do czynienia niemal każdy programista i architekt.

Umiejętności prezentowane w tej książce można by nazwać miękkimi, ale nie jest łatwo się ich nauczyć!

Rebecca Parsons, emerytowana dyrektorka do spraw technologii, Thoughtworks

W książce między innymi:

- techniki przekazywania zamierzonych informacji, odpowiadających danemu etapowi projektu
- archetypy osób pełniących różne funkcje i mających szczególne potrzeby
- techniki komunikacji umożliwiające sukces w środowisku technicznym
- współpraca z rozproszonymi zespołami
- stosowanie wzorców komunikacyjnych w rzeczywistych projektach informatycznych

Jacqui Read jest uznaną na całym świecie architektką rozwiązań. Ma bogate praktyczne doświadczenie w zakresie pisania oprogramowania i tworzenia architektury systemów informatycznych. Jest specjalistką w rozwijaniu praktyk architektonicznych, konstruowaniu architektur ewolucyjnych i wydobywaniu wartości biznesowej z danych.

	KOD KORZYŚCI Sięgnij po więcej! ▶	
 helion.pl	ISBN 978-83-289-1194-9	
 HELION S.A. ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl	 9 788328 911949	
Cena: 69,00 zł		