



Certified Professional for Requirements Engineering

Poziom podstawowy
Syllabus

Stan Bühne
Martin Glinz
Hans van Loenhoud
Stefan Staal

Warunki korzystania

1. Osoby i instytucje szkoleniowe mogą wykorzystywać niniejszy program nauczania (sylabus) jako podstawę do prowadzenia seminariów, pod warunkiem załączenia w materiałach szkoleniowych informacji o prawach autorskich. Od osób korzystających z programu nauczania w celach reklamowych wymagana jest pisemna zgoda IREB e.V..
2. Pojedyncze osoby lub grupy osób mogą wykorzystywać program nauczania (sylabus) jako podstawę dla artykułów, książek lub innych publikacji pochodnych pod warunkiem umieszczenia w publikacjach informacji, że prawa autorskie należą do autorów i ©IREB e.V. jako źródła i właściciela tego dokumentu.

©IREB e.V.

Wszelkie prawa zastrzeżone. Żadna część niniejszej publikacji nie może być powielana, przechowywana w systemie wyszukiwania lub przesyłana w jakiegokolwiek formie za pomocą jakichkolwiek środków elektronicznych, mechanicznych, kserokopii, nagrań lub w inny sposób bez uprzedniej pisemnej zgody autorów lub ©IREB e.V.

Podziękowania

Sylabus został pierwotnie opracowany w 2007 roku, a jego autorami są: Karol Frühauf, Emmerich Fuchs, Martin Glinz, Rainer Grau, Colin Hood, Frank Houdek, Peter Hruschka, Barbara Paech, Klaus Pohl i Chris Rupp. Byli oni wspierani przez: Ian Alexander, Joseph Bruder, Samuel Fricker, Günter Halmans, Peter Jaeschke, Sven Krause, Steffen Lentz, Urte Pautz, Suzanne Robertson, Dirk Schüpferling, Johannes Staub, Thorsten Weyeri oraz Joy Beatty.

Wersja 3 to gruntowny przegląd, którego autorami są: Stan Bühne, Martin Glinz, Hans van Loenhoud i Stefan Staal. Byli oni wspierani przez: Karol Frühauf, Rainer Grau, Kim Lauenroth, Chris Rupp i Camille Salinesi.

W trakcie tego przeglądu uwagi przekazali: Xavier Franch, Karol Frühauf, Rainer Grau, Frank Houdek i Thorsten Weyer. Dodatkowe informacje zwrotne zostały przekazane przez: Wim Decoutere i Hans-Jörg Steffe.

Nową wersję recenzowali: Christoph Ebert, Barbara Paech oraz Chris Rupp.

Dokument został zatwierdzony do publikacji w dniu 22 lipca 2020 r. przez Radę IREB na zlecenie Xavier Franch i Frank Houdek.

Tłumaczenie na język polski zostało opracowane przez SJSI (Stowarzyszenie Jakości Systemów Informatycznych): Radosław Grębski, Joanna Kalabińska (tłumaczenie), Iwona Kolasa (konsultacja merytoryczna) oraz Monika Petri-Starego (redakcja tekstu i korekta tłumaczenia).

Przegląd przeprowadzili: Sebastian Małyska (przegląd i wsparcie), Karolina Zmitrowicz, Jakub Walczak oraz Jakub Bobran.

Przegląd i tłumaczenie poprawek wersji 3.1: Monika Petri-Starego. Iwona Kolasa (konsultacja merytoryczna).

Dziękujemy wszystkim za zaangażowanie.

Prawa autorskie © 2007–2022 do tego sylabusu należą do autorów wymienionych powyżej. Prawa te zostały przeniesione na International Requirements Engineering Board e.V. (IREB), Karlsruhe, Niemcy.

Preambuła

Latem 2017 roku przeprowadziliśmy badanie mające na celu zbadanie przydatności obecnej certyfikacji Certyfikowany specjalista inżynierii wymagań IREB – Poziom podstawowy (IREB Certified Professional for Requirements Engineering – CPRE), wersja 2.2. Celem badania było uzyskanie informacji zwrotnej na temat praktycznego znaczenia rynkowego tej certyfikacji, zarówno z punktu widzenia podmiotów świadczących usługi szkoleniowe, jak i certyfikowanych praktyków CPRE pracujących jako inżynierowie wymagań [MFeA2019]. Badanie wykazało, że obowiązujący program nauczania CPRE poziom podstawowy wersja 2.2 zasadniczo nadal zaspokaja najważniejsze potrzeby rynku i przekazuje kandydatom odpowiednią wiedzę z zakresu inżynierii wymagań. Niemniej jednak otrzymaliśmy informację zwrotną, że kilka technik nie jest już stosowanych w praktyce, podczas gdy innych brakuje, zwłaszcza w odniesieniu do bardziej iteracyjnego i adaptacyjnego procesu wytwarzania. Uzyskane informacje zwrotne były zgodne z postrzeganiem przez IREB zmian w dziedzinie inżynierii wymagań (IW). W związku z tym postanowiliśmy dokonać gruntownego przeglądu programu nauczania CPRE na poziomie podstawowym usuwając nieaktualne treści i dodając nowe elementy. Wersja 3 sylabusu odzwierciedla stan współczesnej inżynierii wymagań, obejmując zarówno sterowane planem, jak i zwinne podejścia do specyfikowania i zarządzania wymaganiami.

Kandydaci ubiegający się o certyfikat CPRE, zgodnie z niniejszym programem nauczania, powinni posiadać podstawową wiedzę na temat wytwarzania systemów w oparciu o podejścia sterowane planem (plan-driven) oraz zwinne (Agile).

Cel dokumentu

Niniejszy sylabus definiuje poziom podstawowy certyfikacji Certyfikowany specjalista inżynierii wymagań IREB (IREB Certified Professional for Requirements Engineering) zdefiniowanej przez International Requirements Engineering Board (IREB). Sylabus zapewnia dostawcom szkoleń podstawę do tworzenia materiałów szkoleniowych. Uczestnicy szkoleń mogą korzystać z sylabusu, aby przygotować się do egzaminu.

Zawartość sylabusu

Poziom podstawowy odpowiada na potrzeby wszystkich osób zajmujących się tematem inżynierii wymagań. Obejmuje to osoby na stanowiskach takich jak: inżynier wymagań, analityk biznesowy, analityk systemowy, właściciel lub kierownik produktu, programista, kierownik projektu, kierownik IT lub ekspert domenowy.

W niniejszym sylabusie i w powiązanim z nim podręczniku dla inżynierii wymagań używany jest skrót IW.

Zakres treści nauczania

Na poziomie podstawowym CPRE przekazywane są podstawowe zasady, które są jednakowo ważne dla różnych systemów (np. aplikacji mobilnych, systemów informacyjnych

lub systemów cybernetycznych). Co więcej, poziom podstawowy CPRE nie zakłada żadnego konkretnego procesu wytwarzania, ani nie jest ukierunkowany na konkretną dziedzinę. Dostawcy szkoleń mogą oferować szkolenia koncentrujące się na konkretnych rodzajach systemów, procesów lub dziedzin, o ile cele edukacyjne niniejszego programu nauczania zostały w pełni uwzględnione.

Poziom szczegółowości sylabusu

Poziom szczegółowości sylabusu umożliwia nauczanie i egzaminowanie w sposób spójny w skali międzynarodowej. Osiągnięcie tego celu jest możliwe dzięki zawarciu w sylabusie następujących treści:

- ogólne cele nauczania
- zawartość wraz z opisem celów nauczania
- referencje do literatury (gdy jest to konieczne).

Cele nauczania / poziomy poznawcze

Wszystkie moduły i cele nauczania w tym sylabusie mają przypisany odpowiedni poziom poznawczy (cognitive level). Sylabus obejmuje następujące poziomy poznawcze:

- **L1: Znać** (opisać, wymienić, scharakteryzować, rozpoznać, nazwać, przypomnieć sobie, ...) – zapamiętać lub odnaleźć wcześniej poznany materiał.
- **L2: Zrozumieć** (wyjaśnić, zinterpretować, uzupełnić, podsumować, uzasadnić, sklasyfikować, porównać, ...) – uchwycić/poznać znaczenie danego materiału lub sytuacji.
- **L3: Zastosować** (specyfikować, pisać, projektować, opracowywać, wdrażać, ...) – stosować wiedzę i umiejętności w danych sytuacjach.

Wyższe poziomy poznawcze obejmują niższe poziomy. Należy pamiętać, że wszystkie terminy ze słownika, które zostały określone jako terminy podstawowe, muszą być znane (L1), nawet jeśli nie są one wyraźnie wymienione w celach nauczania. Słownik jest dostępny do pobrania na stronie głównej IREB pod adresem <https://www.ireb.org/downloads/#cpre-glossary>

Struktura sylabusu

Sylabus składa się z siedmiu rozdziałów. Każdy rozdział obejmuje jedną jednostkę edukacyjną (EU). Tytuły głównych rozdziałów zawierają informację na temat poziomu poznawczego. Jest on określony jako najwyższy poziom poznawczy spośród tych zdefiniowanych dla podrozdziałów. Czas trwania wskazuje jednostkę czasu, jaka powinna być przeznaczona na szkolenie dotyczące danego rozdziału. Firmy szkoleniowe mogą poświęcać więcej czasu danym rozdziałom, ale powinny zachować proporcje pomiędzy poszczególnymi EU. Ważne terminy używane w danym rozdziale są wymienione na początku rozdziału.

Przykład

EU 4 Praktyki opracowywania wymagań (L3)

Czas trwania: 4 godziny 30 minut

Terminy: Źródło wymagań, granica systemu, kontekst systemu, pozyskiwanie wymagań, walidacja wymagań, interesariusz, Model Kano, konflikt

Powyższy przykład wskazuje, że Rozdział 4 zawiera cele nauczania na poziomie L3 oraz zawiera informację, że na nauczanie materiału zawartego w tym rozdziale powinny być przeznaczone cztery i pół godziny.

Każdy rozdział zawiera podrozdziały. Ich tytuły także zawierają poziomy poznawcze odnoszące się do ich treści.

Cele nauczania (EO – Educational Objectives) są wymienione przed właściwą treścią rozdziału. Numeracja wskazuje na podrozdział, do którego one należą. Na przykład cel nauczania EO 4.2.1 został opisany w podrozdziale 4.2.

Kolejność tematów w sylabusie

Kolejność rozdziałów w sylabusie odzwierciedla logiczną kolejność tematów. Tematy nie muszą być jednak realizowane w dokładnie takiej kolejności. Dostawcy szkoleń mogą przekazywać treści w dowolnej kolejności (włączając w to przeplatanie tematów z różnych jednostek edukacyjnych), którą uznają za właściwą w kontekście swojego szkolenia i która odpowiada ich koncepcjom dydaktycznym.

Egzamin

Sylabus jest podstawą egzaminu na certyfikat CPRE na poziomie podstawowym.



Pytania egzaminacyjne mogą obejmować materiał z kilku rozdziałów sylabusu. Pytania egzaminacyjne mogą dotyczyć wszystkich rozdziałów (od EU 1 do EU 7).

Egzamin składa się z pytań wielokrotnego wyboru.

Egzaminy mogą być przeprowadzane bezpośrednio po szkoleniu, ale także niezależnie od szkoleń (np. w centrum egzaminacyjnym). Wykaz licencjonowanych jednostek certyfikujących IREB znajduje się na stronie internetowej <https://www.ireb.org>.

Historia wersji

Wersji	Data	Uwagi
3.0.0	1 grudnia 2021	Zaktualizowany sylabus odzwierciedla stan współczesnej inżynierii wymagań, obejmując zarówno sterowane planem, jak i zwinne podejścia do specyfikowania i zarządzania wymaganiami.
3.1.0	1 listopada 2022	<p>Poprawiono literówki i odniesienia w całym dokumencie, aby poprawić jego czytelność.</p> <p>EU 1:</p> <ul style="list-style-type: none">Przeniesiono cel nauczania z EO 1.3.2 do EO 1.1.2.Zaktualizowano EO 1.2.2 i 1.3.1. <p>EU 3</p> <ul style="list-style-type: none">Zaktualizowano cel nauczania EO 3.1.3UE 3.1.2: Zmieniono akapit dotyczący poziomów abstrakcjiUE 3.4: Wszystkie typy modeli, które nie muszą być stosowane na poziomie podstawowym, zostały przeniesione do nowej sekcji 3.4.6UE 3.6: Zaktualizowano nagłówek "Dokumenty wymagań i struktury dokumentacji" <p>EU 4</p> <ul style="list-style-type: none">UE 4.1: Doprecyzowano opis interesariuszy i ról interesariuszy.UE 4.2: Wprowadzenie i uzasadnienie technik projektowania i generowania pomysłów zostało zaktualizowane, aby było bardziej precyzyjne.UE 4.3: Zaktualizowano nagłówek "Rozwiązywanie konfliktów dotyczących wymagań" <p>EU 6</p> <ul style="list-style-type: none">EO 6.3.1 i EO 6.5.2 zostały lekko zmodyfikowane.
3.1.1	1 stycznia 2024	Nowy design korporacyjny

Spis treści

Spis treści.....	7
1 Wprowadzenie i przegląd inżynierii wymagań (L2).....	10
1.1 Inżynieria wymagań: Co (L1).....	10
1.2 Inżynieria wymagań: Dlaczego (L2).....	11
1.3 Inżynieria wymagań: Gdzie (L2).....	11
1.4 Inżynieria wymagań: Jak (L1).....	11
1.5 Rola i zadania inżyniera wymagań (L1).....	12
1.6 Co warto wiedzieć o inżynierii wymagań (L1).....	12
2 Podstawowe zasady inżynierii wymagań (L2).....	13
2.1 Przegląd zasad (L1).....	13
2.2 Wyjaśnienie zasad (L2).....	13
3 Artefakty i praktyki dokumentowania (L3).....	18
3.1 Artefakty w inżynierii wymagań (L2).....	19
3.1.1 Charakterystyka artefaktów (L1).....	19
3.1.2 Poziomy abstrakcji (L2).....	20
3.1.3 Poziom szczegółowości wymagań (L2).....	20
3.1.4 Aspekty, które należy wziąć pod uwagę w artefaktach (L1).....	21
3.1.5 Ogólne wytyczne dotyczące dokumentacji (L1).....	21
3.1.6 Zaplanowanie artefaktów, które będą wykorzystywane (L1).....	22
3.2 Artefakty oparte na języku naturalnym (L2).....	22
3.3 Artefakty oparte na szablonach (L3).....	23
3.4 Artefakty oparte na modelach (L3).....	24
3.4.1 Rola modeli w inżynierii wymagań (L2).....	24
3.4.2 Modelowanie kontekstu (L2).....	25
3.4.3 Modelowanie struktury i danych (L3).....	25
3.4.4 Modelowanie funkcji i przepływu (L3).....	26

3.4.5	Modelowanie stanu i zachowania [L2]	26
3.4.6	Inne typy modeli w inżynierii wymagań [L1]	26
3.5	Słowniki [L2]	27
3.6	Dokumenty wymagań i struktura dokumentacji [L2]	27
3.7	Prototypy w inżynierii wymagań [L1]	28
3.8	Kryteria jakości artefaktów i wymagań [L1]	29
4	Praktyki w zakresie opracowywania wymagań [L3]	31
4.1	Źródła wymagań [L3]	31
4.2	Pozyskiwanie wymagań [L2]	33
4.3	Rozwiązywanie konfliktów dotyczących wymagań [L2]	34
4.4	Walidacja wymagań [L2]	35
5	Proces i struktura pracy [L3]	37
5.1	Czynniki wpływające [L2]	37
5.2	Aspekty procesu inżynierii wymagań [L2]	38
5.3	Konfigurowanie procesu inżynierii wymagań [L3]	40
6	Praktyki w zakresie zarządzania wymaganiami [L2]	42
6.1	Czym jest zarządzanie wymaganiami? [L1]	42
6.2	Zarządzanie cyklem życia wymagań [L2]	43
6.3	Kontrola wersji [L2]	43
6.4	Konfiguracje i wersje podstawowe (baselines) [L1]	43
6.5	Atrybuty i widoki [L2]	44
6.6	Śledzenie powiązań pomiędzy wymaganiami (traceability) [L1]	44
6.7	Obsługa zmiany [L1]	45
6.8	Priorytetyzacja [L1]	45

7	Narzędzia wspierające (L2)	46
7.1	Narzędzia w inżynierii wymagań (L1)	46
7.2	Wprowadzanie narzędzi (L2)	47
	Odniesienia	48

1 Wprowadzenie i przegląd inżynierii wymagań (L2)

Cel: Poznanie, czym jest inżynieria wymagań i zrozumienie jej wartości

Czas trwania: 1 godzina

Terminy: Wymagania, specyfikacja wymagań, inżynieria wymagań (IW), interesariusz, system, inżynier wymagań

Cele nauczania

- EO 1.1.1 Znajomość podstawowej terminologii (L1)
- EO 1.1.2 Umiejętność rozumienia różnych rodzajów wymagań (L2)
- EO 1.2.1 Umiejętność wyjaśnienia wartości IW (L2)
- EO 1.2.2 Znajomość symptomów i przyczyn niewłaściwej IW (L1)
- EO 1.3.1 Znajomość zastosowań IW i tego, gdzie występują wymagania (L1)
- EO 1.4.1 Znajomość głównych zadań IW oraz konieczności dostosowania procesu IW do ich realizacji (L1)
- EO 1.5.1 Umiejętność scharakteryzowania roli i zadań inżyniera wymagań (L1)
- EO 1.6.1 Znajomość zakresu wiedzy inżyniera wymagań (L1)

1.1 Inżynieria wymagań: Co (L1)

Ludzie i organizacje mają oczekiwania i potrzeby dotyczące budowania nowych lub rozwijania istniejących obiektów (things). Takie potrzeby nazywamy *wymaganiami*.

Obiektami (things), które będą budowane lub rozwijane, mogą być:

- *Produkty* dostarczane klientom
- *Usługi* świadczone na rzecz klientów
- Wszelkie inne *rezultaty (deliverables)* takie jak urządzenia, procedury lub narzędzia, które pomagają ludziom i organizacjom osiągnąć określony *cel*
- *Zestawy* lub *komponenty* produktów, usług lub innych rezultatów

Wszystko to można uznać za *systemy*. W tym sylabusie używamy terminu *system* dla oznaczenia wszystkiego tego, czego wymagają *interesariusze*. *Interesariusze* to osoby lub organizacje, które wpływają na wymagania dotyczące systemu lub osoby i organizacje, na które system ma wpływ.

Celem IW jest specyfikowanie i zarządzanie wymaganiami dla systemów w taki sposób, aby wytwarzane i wdrażane systemy spełniały oczekiwania i potrzeby interesariuszy.

W IW wyróżniamy trzy rodzaje wymagań [Glin2020]:

- *Wymagania funkcjonalne*, które odnoszą się do wyniku, który musi być dostarczony lub zachowania, które musi być zapewnione przez funkcję systemu. Obejmują one wymagania dotyczące danych lub interakcji systemu z jego środowiskiem.
- *Wymagania jakościowe* odnoszą się do zagadnień jakości, które nie są objęte wymaganiami funkcjonalnymi. Są to na przykład: wydajność, dostępność, bezpieczeństwo lub niezawodność.

- *Ograniczenia* to wymagania, które ograniczają przestrzeń rozwiązania ponad to, co jest konieczne, żeby spełnić dane wymagania funkcjonalne i wymagania jakościowe.

1.2 Inżynieria wymagań: Dlaczego (L2)

Właściwa IW wnosi *wartość* w proces wytwarzania i rozwoju systemu:

- zmniejsza ryzyko wytworzenia niewłaściwego systemu
- umożliwia lepsze zrozumienie problemu
- jest podstawą do oszacowania wysiłku i kosztu wytwarzania
- zapewnia warunki wstępne testowania systemu.

Typowe symptomy niewłaściwej IW to brakujące, niejasne lub nieprawidłowe wymagania.

Jest to spowodowane w szczególności przez:

- pospieszne budowanie systemu
- problemy komunikacyjne między zaangażowanymi stronami
- założenie, że wymagania są oczywiste
- niewystarczającą edukację i umiejętności w zakresie IW.

1.3 Inżynieria wymagań: Gdzie (L2)

IW ma zastosowanie dla systemu dowolnego rodzaju. Jednakże obecnie dominującym przypadkiem zastosowania IW są systemy, w których główną rolę odgrywa oprogramowanie. Takie systemy zazwyczaj składają się z elementów oprogramowania, elementów fizycznych i organizacyjnych.

Wymagania mogą występować jako:

- *wymagania systemowe* — określają, co system powinien robić
- *wymagania interesariuszy* — określają, czego interesariusze chcą ze swojej perspektywy
- *wymagania użytkowników* — określają, czego użytkownicy chcą ze swojej perspektywy
- *wymagania dotyczące domeny* — definiują wymagane właściwości domeny
- *wymagania biznesowe* — określają cele biznesowe, założenia i potrzeby organizacji.

1.4 Inżynieria wymagań: Jak (L1)

Głównymi zadaniami w IW są: pozyskiwanie (4.2), dokumentacja (3), walidacja (4.4) oraz zarządzanie (6) wymaganiami. W realizacji tych zadań mogą pomóc narzędzia wspierające (7). Analiza wymagań i rozwiązywanie konfliktów dotyczących wymagań (4.3) są uważane za część pozyskiwania wymagań. Aby poprawnie wykonać czynności związane z IW, z szerokiego zakresu możliwości należy dopasować (dobrać) odpowiedni proces IW (5).

1.5 Rola i zadania inżyniera wymagań (L1)

Inżynier wymagań zazwyczaj nie jest nazwą stanowiska, ale *rolą* realizowaną przez osoby, które:

- w ramach swoich obowiązków pozyskują, dokumentują, walidują i/lub zarządzają wymaganiami
- posiadają dogłębną wiedzę na temat IW
- mogą wypełniać lukę pomiędzy problemem a potencjalnymi rozwiązaniami.

W praktyce analitycy biznesowi, specjaliści ds. aplikacji, właściciele produktów, inżynierowie systemów, a nawet programiści pełnią rolę inżynierów wymagań.

1.6 Co warto wiedzieć o inżynierii wymagań (L1)

Sylabus obejmuje podstawowy zestaw umiejętności, jakich musi nauczyć się inżynier wymagań. Obejmuje on podstawowe zasady IW (2), wskazówki, jak dokumentować wymagania w różnych formach (3), jak opracowywać wymagania przy użyciu różnych technik (4), jak definiować i pracować z odpowiednimi procesami IW (5), jak zarządzać istniejącymi wymaganiami (6) oraz jak wybrać i wdrożyć narzędzia wspierające (7).

2 Podstawowe zasady inżynierii wymagań (L2)

Cel: Poznanie i zrozumienie zasad IW

Czas trwania: 1 godzina 30 minut

Terminy: Kontekst, wymaganie, inżynieria wymagań (IW), interesariusz, wspólne zrozumienie, walidacja

Cele nauczania

EO 2.1.1 Znajomość zasad IW (L1)

EO 2.2.1 Znajomość terminów związanych z zasadami IW (L1)

EO 2.2.2 Umiejętność wyjaśnienia zasad IW i przyczyn, dla których są one ważne (L2)

2.1 Przegląd zasad (L1)

IW podlega zestawowi podstawowych zasad, które mają zastosowanie do wszystkich zadań, działań i praktyk w ramach inżynierii wymagań. Poniższe dziewięć zasad stanowi podstawę dla praktyk przedstawionych w kolejnych rozdziałach niniejszego programu nauczania.

1. Orientacja na wartość: wymagania są środkiem do osiągnięcia celu, a nie celem samym w sobie.
2. Interesariusze: IW polega na zaspokajaniu oczekiwań i potrzeb interesariuszy.
3. Wspólne zrozumienie: efektywny rozwój systemów nie jest możliwy bez wspólnego zrozumienia.
4. Kontekst: systemy nie mogą być zrozumiane w izolacji.
5. Problem – Wymaganie – Rozwiązanie: nierozłączna trójka (an inevitably intertwined triple)
6. Walidacja: niezwalidowane wymagania są bezużyteczne.
7. Ewolucja: zmieniające się wymagania nie są wyjątkiem, a regułą.
8. Innowacja: więcej tego samego nie wystarczy.
9. Systematyczna i zdyscyplinowana praca: nie możemy się obejść bez IW.

2.2 Wyjaśnienie zasad (L2)

Zasada 1 - Orientacja na wartość: wymagania są środkiem do osiągnięcia celu, a nie celem samym w sobie

Wartość wymagania jest równa korzyściom z nim związanym pomniejszonym o koszty pozyskania, dokumentowania, walidacji i zarządzania wymaganiem. Korzyścią związaną z wymaganiem jest stopień, w jakim przyczynia się ono do:

- budowania systemów, które zaspokajają oczekiwania i potrzeby ich interesariuszy
- zmniejszenia ryzyka awarii i kosztownych zmian podczas wytwarzania systemu.

Zasada 2 - Interesariusze: IW polega na zaspokajaniu oczekiwań i potrzeb interesariuszy

Ponieważ IW polega na zrozumieniu oczekiwań i potrzeb interesariuszy, właściwe postępowanie z interesariuszami jest podstawowym zadaniem inżynierii wymagań. Każdy interesariusz ma przypisaną rolę w kontekście systemu, który ma zostać zbudowany – na przykład użytkownika, klienta, operatora lub regulatora. Interesariusz może również pełnić więcej niż jedną rolę w tym samym czasie. Gdy występuje wielu pojedynczych interesariuszy, którzy tworzą grupę, lub gdy osoby pełniące daną rolę są nieznane, można utworzyć fikcyjne archetypowe opisy (postaci) nazywane *personami*. Nie wystarczy wziąć pod uwagę jedynie wymagań użytkowników końcowych lub klientów. Oznaczałoby to możliwość pominięcia krytycznych wymagań innych interesariuszy. Użytkownicy przekazujący informacje zwrotne na temat użytkowanego systemu również powinni być traktowani jak interesariusze.

Interesariusze mogą mieć różne potrzeby i punkty widzenia, co może prowadzić do sprzecznych wymagań. Zadaniem IW jest identyfikowanie i rozwiązywanie takich konfliktów.

Zaangażowanie właściwych osób jako interesariuszy o odpowiednich rolach jest kluczowe dla efektywnej IW. Praktyki w zakresie identyfikowania, priorytetyzacji i pracy z interesariuszami zostały opisane w 4.

Zasada 3 - Wspólne zrozumienie: efektywny rozwój systemów nie jest możliwy bez wspólnego zrozumienia.

IW tworzy, wspiera i zapewnia wspólne zrozumienie pomiędzy zaangażowanymi stronami: interesariuszami, inżynierami wymagań a deweloperami. Istnieją dwie formy wspólnego zrozumienia:

- *Faktyczne (explicit) wspólne zrozumienie* (osiągnięte poprzez udokumentowane i uzgodnione wymagania).
- *Domniemane (implicit) wspólne zrozumienie* (oparte na wspólnej wiedzy o potrzebach, wizjach, kontekście itp.).

Wiedza domenowa, wcześniejsza efektywna współpraca, wspólna kultura i wartości oraz wzajemne zaufanie są czynnikami sprzyjającymi wspólnemu zrozumieniu, natomiast odległość geograficzna, outsourcing lub duże zespoły o wysokiej rotacji są przeszkodami.

Sprawdzone praktyki osiągnięcia wspólnego zrozumienia obejmują tworzenie słowników (3.5), tworzenie prototypów (3.7) lub wykorzystywanie istniejącego systemu jako punktu odniesienia. Praktyki oceny wspólnego zrozumienia obejmują omawianie przykładów oczekiwanych wyników, badanie prototypów lub szacowanie kosztów wdrożenia wymagania. Najważniejszą praktyką służącą ograniczeniu wpływu nieporozumień jest stosowanie procesu z krótkimi pętlami informacji zwrotnej (5).

Zasada 4 - Kontekst: systemy nie mogą być zrozumiane w izolacji

Systemy są osadzone w *kontekście*. Bez zrozumienia tego kontekstu nie jest możliwe prawidłowe określenie systemu. W IW kontekst systemu jest tą częścią środowiska systemu, która jest istotna dla zrozumienia systemu i dotyczących go wymagań. *Granica systemu* to granica pomiędzy systemem a otaczającym go kontekstem. Początkowo granica systemu często jest niejasna, a z czasem może nawet ulegać zmianie.

Sprecyzowanie granicy systemu i zdefiniowanie zewnętrznych interfejsów pomiędzy systemem a elementami kontekstu, z którymi system wchodzi w interakcję, należy do kluczowych zadań IW. Jednocześnie określenia wymaga zakres systemu — czyli zakres obiektów (*things*), które mogą być kształtowane i projektowane podczas wytwarzania systemu. Musimy również wziąć pod uwagę tak zwaną *granicę kontekstu*, która oddziela istotną dla IW część środowiska systemu od reszty świata.

Przy określaniu systemu nie wystarczy wziąć pod uwagę wyłącznie wymagań (znajdujących się) w obrębie granicy systemu. W procesie IW trzeba również rozważyć:

- Zmiany w kontekście, które mogą wpływać na wymagania dotyczące systemu.
- Wymagania dotyczące świata rzeczywistego istotne dla systemu (i sposób ich mapowania na wymagania systemowe).
- Założenia dotyczące kontekstu, które muszą być spełnione, aby system działał i spełniał wymagania świata rzeczywistego.

Zasada 5 - Problem - Wymaganie - Rozwiązanie: połączony trójkąt (an inevitably intertwined triple)

Problem pojawia się, gdy interesariusze nie są zadowoleni ze stanu obecnego. *Wymagania* definiują to, czego interesariusze potrzebują, aby pozbyć się problemu lub go uprościć. *Rozwiązaniem* jest system socjotechniczny, który spełnia zdefiniowane wymagania.

Problemy, wymagania i rozwiązania nie muszą występować w wymienionej kolejności. Pomysły dotyczące rozwiązań mogą tworzyć potrzeby użytkowników, które muszą być opracowane jako wymagania i wdrożone w rzeczywistym rozwiązaniu. Jest to typowe w przypadku wprowadzania innowacji.

- Problemy, wymagania i rozwiązania są ściśle ze sobą powiązane: nie można ich rozpatrywać w izolacji.
- Niemniej jednak inżynierowie wymagań dążą do oddzielenia od siebie problemów, wymagań i rozwiązań tak dalece, jak to możliwe, podczas myślenia, komunikowania się i dokumentowania. To rozdzielenie kwestii sprawia, że zadania związane z IW są łatwiejsze do realizacji.

Zasada 6 - Walidacja: niezwalidowane wymagania są bezużyteczne

Ostatecznie musimy zwalidować, czy wdrożony system spełnia oczekiwania i potrzeby interesariuszy. W celu kontroli ryzyka niezadowolenia interesariuszy od samego początku, walidacja wymagań musi się rozpocząć już w trakcie procesu IW. Musimy sprawdzić, czy:

- porozumienie dotyczące wymagań zostało osiągnięte przez interesariuszy
- oczekiwania i potrzeby interesariuszy są odpowiednio uwzględnione w wymaganiach
- założenia dotyczące kontekstu (patrz Zasada 4 powyżej) są rozsądne.

Praktyki w zakresie walidacji wymagań zostały omówione w 4.4.

Zasada 7 - Ewolucja: zmiana wymagań nie jest wyjątkiem, a normą

Systemy i ich wymagania podlegają *ewolucji*. Oznacza to, że ciągle się zmieniają. Żądania zmian dotyczące wymagania lub zestawu wymagań dla systemu mogą być spowodowane na przykład przez:

- zmienione procesy biznesowe
- konkurentów wprowadzających na rynek nowe produkty lub usługi
- klientów zmieniających swoje priorytety lub opinie
- zmiany technologiczne
- zmiany przepisów i regulacji prawnych
- informacje zwrotne od użytkowników systemu zgłaszających nowe lub zmienione funkcjonalności.

Ponadto wymagania mogą się zmienić w związku z informacjami zwrotnymi uzyskanymi od interesariuszy podczas walidowania wymagań, z powodu wykrycia wad we wcześniej pozyskanych wymaganiach lub z powodu zmiany potrzeb interesariuszy.

W konsekwencji inżynierowie wymagań muszą dążyć do dwóch pozornie sprzecznych celów:

- zezwalać na zmiany wymagań
- dbać o stabilność wymagań.

Szczegóły dotyczące tego, jak to osiągnąć, zostały omówione w 6.7.

Zasada 8 - Innowacje: więcej tego samego nie wystarcza

Dając interesariuszom dokładnie to, czego chcą, tracimy szansę na zbudowanie systemów, które zaspokajają ich potrzeby lepiej niż interesariusze by się tego spodziewali. Właściwa IW dąży nie tylko do usatysfakcjonowania interesariuszy, ale także do tego, aby byli szczęśliwi, podekscytowani lub czuli się bezpiecznie. Taki jest ostateczny cel innowacji.

IW umożliwia kształtowanie innowacyjnych systemów:

- na małą skalę poprzez dążenie do nowych, ekscytujących funkcjonalności i łatwości ich obsługi
- na dużą skalę poprzez dążenie do wdrażania nowych, przełomowych pomysłów.

W 4.2 omówiono kilka technik wspierania innowacji w IW.

Zasada 9 – Systematyczna i zdyscyplinowana praca: nie możemy się obejść bez IW

Konieczne jest stosowanie odpowiednich procesów i praktyk w zakresie systematycznego pozyskiwania, dokumentowania, walidowania i zarządzania wymaganiami, niezależnie od stosowanego procesu wytwarzania. Nawet wtedy, gdy system jest wytwarzany ad-hoc, systematyczne i zdyscyplinowane podejście do IW poprawia jakość wynikowego systemu.

W IW nie istnieje jeden proces lub praktyka, które sprawdzają się w każdej sytuacji lub przynajmniej w większości sytuacji. Systematyczna i zdyscyplinowana praca oznacza, że inżynierowie wymagają:

- Dostosowują swoje procesy i praktyki do danego problemu, kontekstu i środowiska.
- Nie zawsze stosują ten sam proces i zestaw praktyk.
- Nie wykorzystują ponownie procesów i praktyk z poprzednich udanych projektów IW bez refleksji.

Dla każdego przedsięwzięcia dotyczącego inżynierii wymagań procesy, praktyki i artefakty muszą być wybierane tak, aby były jak najlepiej dopasowane do konkretnej sytuacji. Szczegóły zostały opisane w 3, 4, 5 i 6.

3 Artefakty i praktyki dokumentowania (L3)

Cel: Zrozumienie fundamentalnej roli artefaktów w IW i możliwość tworzenia artefaktów

Czas trwania: 7 godzin

Terminy: Artefakty, artefakty oparte na języku naturalnym, artefakty oparte na szablonach, artefakty oparte na modelach, słownik, kryteria jakości, specyfikacja wymagań

Cele nauczania

- EO 3.1.1 Znajomość cech charakterystycznych artefaktów IW i umiejętność wskazania często używanych typów artefaktów (L1)
- EO 3.1.2 Znajomość przeznaczenia każdego artefaktu oraz znajomość długości życia artefaktów (L1)
- EO 3.1.3 Umiejętność scharakteryzowania różnych poziomów abstrakcji wymagań, w tym sposobu wyboru odpowiednich poziomów abstrakcji i poziomów szczegółowości wymagań (L2)
- EO 3.1.4 Znajomość aspektów, które należy uwzględnić w procesie tworzenia artefaktów oraz wzajemnych powiązań między tymi aspektami (L1)
- EO 3.1.5 Umiejętność nazwania ogólnych wytycznych dotyczących dokumentacji (L1)
- EO 3.1.6 Umiejętność opisanie, dlaczego warto zaplanować, które artefakty mają być wykorzystane (L1)
- EO 3.2.1 Znajomość artefaktów bazujących na języku naturalnym oraz ich zalet i wad (L1)
- EO 3.2.2 Umiejętność wyjaśnienia zasad pisania wysokiej jakości wymagań z wykorzystaniem języka naturalnego (L2)
- EO 3.3.1 Znajomość kategorii artefaktów opartych na szablonach oraz ich zalet i wad (L1)
- EO 3.3.2 Umiejętność specyfikowania pojedynczego wymagania i historyjki użytkownika przy wykorzystaniu szablonu wyrażenia (phrase template) (L3)
- EO 3.3.3 Umiejętność specyfikowania przypadku użycia za pomocą szablonu (L3)
- EO 3.4.1 Zrozumienie roli, celu i wykorzystania modeli w IW (L2)
- EO 3.4.2 Zrozumienie zalet i ograniczeń modelowania w IW (L2)
- EO 3.4.3 Znajomość pojęć: model, język modelowania, model aktywności, diagram aktywności, model klas, diagram klas, diagram kontekstowy, model domenowy, model celów, model interakcji, model procesu, diagram sekwencji, schemat stanów, maszyna stanów, diagram maszyny stanów, przypadek użycia, diagram przypadków użycia (L1)
- EO 3.4.4 Zrozumienie, w jaki sposób dokonać wyboru odpowiedniego typu modelu do wyspecyfikowania wymagań w danej sytuacji (L2)
- EO 3.4.5 Umiejętność interpretacji i zrozumienie prostych modeli napisanych w języku UML tam, gdzie ma to zastosowanie, na przykład: modeli kontekstowych, przypadków użycia i diagramów przypadków użycia, modeli domeny, modeli klas, modeli aktywności, modeli procesów i schematów stanów (statecharts) (L2)
- EO 3.4.6 Umiejętność wyspecyfikowania prostych modeli danych systemu lub obiektów w domenie za pomocą diagramu klas UML (L3)
- EO 3.4.7 Umiejętność wyspecyfikowania prostej funkcji systemu lub procesu biznesowego za pomocą diagramu aktywności UML (L3)
- EO 3.5.1 Umiejętność wyjaśnienia celu słowników i sposobu tworzenia słownika (L2)
- EO 3.6.1 Znajomość często stosowanych rodzajów dokumentów specyfikacji wymagań (L1)
- EO 3.6.2 Umiejętność wyjaśnienia celów poszczególnych struktur dokumentów i kryteriów ich konstruowania (L2)

- EO 3.7.1 Znajomość różnych rodzajów prototypów oraz tego, do czego są one wykorzystywane (L1)
- EO 3.8.1 Znajomość kryteriów jakości dla pojedynczych wymagań (L1)
- EO 3.8.2 Znajomość kryteriów jakości dla artefaktów (L1).

3.1 Artefakty w inżynierii wymagań (L2)

Artefakt to zarejestrowany, pośredni lub końcowy, rezultat wytworzony w procesie pracy. W IW wyróżnić można wiele różnych artefaktów, począwszy od tymczasowych szkiców (short-lived graphic sketches), poprzez stale rozwijane zbiory historyjek użytkownika (user stories), aż po formalne, liczące setki stron, dokumenty specyfikacji wymagań.

3.1.1 Charakterystyka artefaktów (L1)

Artefakty charakteryzują się przeznaczeniem, formą, wielkością i długością życia (obowiązywania). Poniższe artefakty są często stosowane w praktyce w określonym celu. Należy pamiętać, że dany artefakt może zawierać inne artefakty.

- Artefakty dla pojedynczego wymagania obejmują pojedyncze wymagania i historyjki użytkownika (user stories).
- Artefakty dla spójnego zestawu wymagań obejmują przypadki użycia, modele graficzne danego typu (3.4), opisy zadań, opisy interfejsów zewnętrznych i epiki (epics).
- Artefakty, które stanowią obszerne (wyczerpujące) dokumenty lub struktury dokumentacji, obejmują specyfikacje wymagań systemowych, rejestry produktów (product backlogs) i sprintów (sprint backlogs) oraz mapy historyjek użytkownika (story maps).
- Inne artefakty obejmują słowniki, notatki tekstowe, szkice graficzne i prototypy.

Artefakty mogą być *przedstawiane* w różnych formach:

- oparte na języku naturalnym (3.2)
- oparte na szablonie (3.3)
- oparte na modelu (3.4)
- inne reprezentacje, takie jak rysunki lub prototypy (3.7).

Większość artefaktów jest *przechowywana* elektronicznie w postaci plików, w bazach danych lub w narzędziach IW. Nieformalne, tymczasowe artefakty, mogą być również przechowywane na innych nośnikach — na przykład na papierze, jako notatki post-it lub na tablicy Kanban.

Patrząc na długość życia artefaktów rozróżniamy trzy kategorie:

- *Tymczasowe artefakty*: stworzone w celu wspierania komunikacji i tworzenia wspólnego zrozumienia.
- *Rozwijane artefakty*: zazwyczaj pojawiają się w trakcie kilku iteracji, mogą wykorzystywać metadane (6.5), mogą podlegać kontroli zmian.

- *Trwałe artefakty*: zostały uznane za podstawowe (*baselined*) lub zostały dostarczone (*released*), wymagają pełnego zestawu metadanych (6.5), muszą podlegać procesowi zmian (6.3, 6.4).

Artefakt tymczasowy może zostać przekształcony w artefakt rozwijany (poprzez utrzymywanie go i dodawanie metadanych). Analogicznie, tymczasowy lub rozwijany artefakt może stać się trwałym artefaktem dzięki temu, że zostanie on uznany za podstawowy (*baselined*) lub dostarczony (*released*).

3.1.2 Poziomy abstrakcji (L2)

Wymagania istnieją zazwyczaj na wielu *różnych poziomach abstrakcji* — np. od wymagań wysokopoziomowych dla nowego procesu biznesowego, aż do wymagań na bardzo szczegółowym poziomie takich jak reakcja konkretnego komponentu oprogramowania na zdarzenie wyjątkowe.

Wybór właściwego poziomu abstrakcji zależy od przedmiotu specyfikacji oraz od celu specyfikacji. Ważne jest jednak, aby nie zestawiać ze sobą wymagań, które są na różnych poziomach abstrakcji. W przypadku małych i średnich artefaktów, wymagania powinny być na mniej więcej tym samym poziomie abstrakcji.

W przypadku dużych artefaktów takich jak specyfikacja wymagań systemowych, wymagania na różnych poziomach abstrakcji, powinny być one przechowywane oddzielnie za pomocą odpowiedniej struktury dokumentu specyfikacji (3.6). Wymaganie na wysokim poziomie abstrakcji może być dopracowane w kilku szczegółowych wymaganiach na niższych, bardziej konkretnych poziomach.

Kiedy wysokopoziomowe wymagania biznesowe lub interesariuszy są wyrażone w trwałych artefaktach takich jak: specyfikacje wymagań biznesowych, specyfikacje wymagań interesariuszy lub w dokumencie opisującym wizję, wówczas poprzedzają one specyfikację wymagań systemowych. W innych okolicznościach wymagania biznesowe, wymagania interesariuszy i wymagania systemowe mogą być rozwijane wspólnie.

3.1.3 Poziom szczegółowości wymagań (L2)

Poziom *szczegółowości*, do którego wymagania powinny być określone, zależy od kilku czynników, a w szczególności od:

- problemu i kontekstu wytwarzania
- stopnia wspólnego rozumienia problemu
- stopnia swobody pozostawionej projektantom i programistom
- dostępności szybkiej informacji zwrotnej od interesariuszy podczas projektowania i wdrażania
- kosztu w relacji do wartości szczegółowej specyfikacji
- nałożonych norm i ograniczeń regulacyjnych.

Im wyższy poziom szczegółowości opisu wymagań, tym niższe ryzyko otrzymania na końcu czegoś niespodziewanego lub nieokreślonego. Koszt specyfikacji wzrasta jednak wraz ze wzrostem poziomu szczegółowości.

3.1.4 Aspekty, które należy wziąć pod uwagę w artefaktach (L1)

Przy określaniu wymagań za pomocą artefaktów należy wziąć pod uwagę różne aspekty.

1. Wymagania są klasyfikowane według ich rodzaju (1.1) jako:
 - a) wymagania funkcjonalne
 - b) wymagania jakościowe
 - c) ograniczenia
2. Wymagania funkcjonalne koncentrują się na różnych aspektach funkcjonalności systemu:
 - a) strukturze i danych
 - b) funkcji i przepływie
3. stanie i zachowaniu
 - a) Ostatecznie wymagania mogą być zrozumiane tylko w kontekście (Zasada 4 w 2):
 - b) *kontekst systemu*, włączając zewnętrznych aktorów
 - c) *granica systemu* i interfejsy zewnętrzne.

Istnieje wiele wzajemnych powiązań i zależności pomiędzy wymienionymi aspektami. Na przykład żądanie zgłoszone przez użytkownika (kontekst) może wywołać zmianę stanu (stan i zachowanie), która inicjuje działanie, po którym następuje inne działanie (funkcja i przepływ), które wymaga danych (struktura i dane), by dostarczyć użytkownikowi (kontekst) wynik w określonym przedziale czasu (jakość).

Niektóre artefakty koncentrują się na konkretnym aspekcie i odciągają od innych aspektów. Zdarza się tak w szczególności w przypadku modeli wymagań (3.4). Inne artefakty takie jak specyfikacja wymagań systemowych obejmują wszystkie te aspekty. Kiedy różne aspekty są udokumentowane w oddzielnych artefaktach lub w oddzielnych częściach tego samego artefaktu, to te artefakty muszą być ze sobą spójne.

3.1.5 Ogólne wytyczne dotyczące dokumentacji (L1)

Niezależnie od zastosowanych technik, przy tworzeniu artefaktów mają zastosowanie następujące wytyczne:

- Wybieraj typ artefaktu, który odpowiada *zamierzonemu celowi*.
- *Unikaj nadmiarowości* poprzez odwoływanie się do treści zamiast ponownego powtarzania tej samej treści.
- *Upewnij się, że nie ma niespójności* pomiędzy artefaktami, szczególnie wtedy, gdy obejmują one różne aspekty.
- *Używaj terminów w sposób spójny*, zgodnie z ich definicją w słowniku.
- Odpowiednio *strukturyzuj* artefakty.

3.1.6 Zaplanowanie artefaktów, które będą wykorzystywane (L1)

Każdy projekt i każda dziedzina jest inna, więc dla każdego przedsięwzięcia należy zdefiniować zestaw artefaktów. W związku z tym należy uzgodnić następujące kwestie:

- W których artefaktach wymagania muszą zostać udokumentowane i w jakim celu?
- Jakiego poziomu abstrakcji należy wziąć pod uwagę?
- Do jakiego poziomu szczegółowości wymagania muszą być udokumentowane na każdym poziomie abstrakcji?
- W jaki sposób wymagania powinny być przedstawione w określonych artefaktach?

Artefakty, które mają być wykorzystane, powinny być określone na wczesnym etapie projektu. Ma to kilka zalet takich jak:

- Pomaga w planowaniu czynności i zasobów.
- Zapewnia zastosowanie odpowiednich notacji.
- Zapewnia, że wszystkie wyniki są zapisywane we właściwych artefaktach.
- Gwarantuje brak konieczności dokonywania większych zmian informacji i „ostatecznej redakcji”.
- Pomaga uniknąć nadmiarowości, co przekłada się na mniejszą ilość pracy i łatwiejsze utrzymanie.

3.2 Artefakty oparte na języku naturalnym (L2)

Od początku istnienia systematycznej IW język naturalny jest podstawowym środkiem określania wymagań.

Artefakty oparte na języku naturalnym mają wiele zalet:

- Nieograniczony język naturalny jest niezwykle wyrazisty i elastyczny.
- Prawie każde wymaganie, w każdym aspekcie, może być wyrażone w języku naturalnym.
- Język naturalny jest używany w życiu codziennym i nauczany w szkole, więc żadne szkolenie nie jest wymagane, aby czytać i rozumieć teksty w języku naturalnym.

Jednakże powyższe zalety czasami stają się wadami, gdyż teksty napisane w języku naturalnym mogą być często interpretowane na różne sposoby, co stanowi problem przy określaniu wymagań. Ponadto wykrycie niejasności, pominięć i niespójności w takich tekstach jest trudne i kosztowne.

Pisanie wysokiej jakości wymagań przy wykorzystaniu języka naturalnego może być wspierane przez zasady:

- Pisanie krótkich i poprawnie skonstruowanych zdań.
- Definiowanie i systematyczne stosowanie jednolitej terminologii (3.5).
- Unikanie niejasnych lub niejednoznacznych terminów i zwrotów.
- Znajomość pułapek języka technicznego wymienionych poniżej.

Podczas pisania dokumentów technicznych w języku naturalnym istnieją pewne dobrze znane pułapki, których należy unikać lub stosować je ostrożnie [GoRu2003].

Elementy języka naturalnego, których należy unikać w dokumentacji:

- niekompletne opisy
- nieokreślone rzeczowniki
- niekompletne warunki
- niekompletne porównania.

Elementy języka naturalnego i konstrukcje, których należy używać ostrożnie:

- strona bierna
- uniwersalne kwantyfikatory (takie jak: „wszystkie” lub „nigdy”)
- nominalizacje (np. rzeczowniki pochodne od czasownika, np. „uwierzytelnienie”).

3.3 Artefakty oparte na szablonach (L3)

Artefakty wykorzystujące szablony są stosowane w celu uniknięcia niektórych wad charakterystycznych dla artefaktów pisanych w języku naturalnym, poprzez zapewnienie z góry określonych struktur wymagań.

- *Szablony wyrażen* (fraz – *phrase template*) zapewniają predefiniowaną strukturę składniową dla danej frazy, która wyraża wymaganie, w szczególności pojedyncze wymaganie lub historijkę użytkownika (user story).
- *Szablony formularzy* zawierają zestaw predefiniowanych pól w formularzu, które należy wypełnić, na przykład w celu napisania przypadku użycia lub mierzalnego wymagania jakościowego.
- *Szablony dokumentów* zapewniają predefiniowaną strukturę dokumentu wymagań.

W literaturze zostały opisane różne szablony. [ISO29148], [MWHN2009] i [Rupp2014] dostarczają szablony wyrażen dla wymagań indywidualnych. [Cohn 2004] definiuje szeroko stosowany szablon wyrażen dla historyjek użytkownika i [Cock2001] opisuje szablony formularzy dla przypadków użycia. [Laue2002] zaproponował szablon do opisu zadań. [ISO29148] i [RoRo2012] dostarczają wzory dokumentów dla całej specyfikacji. Ponadto klient może narzucać użycie w projekcie specyficznych dla jego organizacji szablonów.

Zalety artefaktów wykorzystujących szablony:

- zapewniają przejrzystą strukturę nadającą się do ponownego wykorzystania
- pomagają w wychyceniu najistotniejszych informacji
- sprawiają, że wymagania i specyfikacje wymagań wyglądają jednolicie
- poprawiają ogólną jakość wymagań i specyfikacji wymagań.

Wady i pułapki artefaktów wykorzystujących szablony:

- ludzie często skupiają się na formalnym wypełnieniu szablonu, a nie na treści
- aspekty, które nie są zawarte w szablonie, są często pomijane.

3.4 Artefakty oparte na modelach (L3)

Wymagania udokumentowane w języku naturalnym mają ograniczenia [Davi1993], w szczególności w odniesieniu do uzyskania ogólnego zrozumienia zbioru wymagań i zrozumienia zależności między nimi. Modelowanie wymagań niweluje te ograniczenia.

3.4.1 Rola modeli w inżynierii wymagań (L2)

Model jest streszczoną reprezentacją istniejącej części rzeczywistości lub części rzeczywistości, która ma zostać stworzona. Pojęcie rzeczywistości obejmuje każdy możliwy zestaw elementów, zjawisk lub koncepcji, w tym inne modele. W odniesieniu do modelu, modelowana część rzeczywistości nazywana jest *oryginałem*. Przykładem modeli spoza dziedziny inżynierii oprogramowania są modele informacji o budynku (BIM – building information models) [ISO19650], które modelują elementy wymagane do planowania, budowania i zarządzania budynkami i innymi elementami konstrukcyjnymi.

W IW modele pomagają zrozumieć wzajemne powiązania pomiędzy wymaganiami oraz prezentują ogólny obraz zbioru wymagań. Osiąga się to przede wszystkim poprzez skupienie się na niektórych aspektach, jednocześnie abstrahując od wszystkich innych aspektów. Wysokopoziomowy obraz wymagań jest również wspierany przez użycie graficznej notacji dla modelu. Jednakże modele mogą być również przedstawiane w sposób inny niż graficzny, na przykład za pomocą tabel.

Modele wymagań, w odróżnieniu od wymagań udokumentowanych w języku naturalnym, mają następujące zalety:

- Zależności i powiązania między wymaganiami są łatwiejsze do zrozumienia przy użyciu modeli graficznych niż przy określaniu ich w języku naturalnym.
- Skupienie się na jednym aspekcie zmniejsza ładunek poznawczy (cognitive load) i ułatwia zrozumienie modelowanych wymagań.
- Języki modelowania wymagań mają sprecyzowaną składnię, co zmniejsza ewentualne niejasności i luki (w zakresie kompletności wymagań).

Modele mają również ograniczenia:

- Utrzymanie spójności modeli, które skupiają się na różnych aspektach, stanowi wyzwanie.
- Dla uzyskania poprawnego zrozumienia wymagań konieczne jest połączenie informacji zawartych w modelach różnego typu.
- Modele koncentrują się przede wszystkim na wymaganiach funkcjonalnych; większość wymagań jakościowych i ograniczeń nie może być wyrażona w modelach w efektywny sposób.
- Ograniczona składnia języka modelowania graficznego implikuje to, że nie każdy istotny element informacji może być wyrażony w modelu.

Dlatego też modele wymagań i wymagania w języku naturalnym są często łączone.

W IW modele mogą być używane do:

- *Specyfikowania* (głównie funkcjonalnych) wymagań w części lub nawet w całości jako sposób na zastąpienie wymagań przedstawionych słownie.
- *Dekompozycji* złożonej rzeczywistości na dobrze zdefiniowane i uzupełniające się aspekty; każdy aspekt jest przedstawiony w konkretnym modelu.
- *Parafrazowania* wymagań przedstawionych słownie w celu poprawienia stopnia ich zrozumiałości, w szczególności w odniesieniu do wzajemnych relacji między nimi.
- *Walidowania* wymagań przedstawionych słownie w celu wykrycia luk, niejednoznaczności i niespójności.

Do tworzenia modeli używane są *języki modelowania*. Kilka języków modelowania, na przykład UML [OMG2017] lub BPMN [OMG2013], zostało znormalizowanych. Gdy wymagania są określone w niestandardowym języku modelowania, wymagana jest legenda wyjaśniająca składnię i semantykę zastosowanego języka modelowania.

Istnieje wiele typów modeli, które mogą być używane w IW. Inżynier wymagań musi zrozumieć, który typ modelu jest najbardziej odpowiedni do wyspecyfikowania wymagań w danej sytuacji.

We wczesnej fazie inżynier wymagań często zaczyna od modelowania kontekstu (3.4.2) lub celów zamierzonego systemu.

3.4.2 Modelowanie kontekstu (L2)

Modele, które skupiają się na aspekcie kontekstu, określają strukturalne osadzenie systemu w jego środowisku oraz interakcję między systemem a aktorami w kontekście systemu.

Modele kontekstowe określają system i aktorów, którzy wchodzi w interakcję z systemem, w kontekście systemu. Model kontekstowy definiuje również interfejsy pomiędzy systemem a jego kontekstem (np. w zakresie tego, jakie informacje są wymieniane).

Do prezentacji graficznej modeli kontekstowych używa się *diagramów kontekstowych* jako graficznego języka modelowania. Nie ma standardowej notacji dla diagramów kontekstowych. Diagramy kontekstowe z analizy strukturalnej [DeMa1978] lub diagramy używające bloków i linii [Glin2019] mogą być używane przy tworzeniu modeli kontekstowych.

W języku modelowania UML [OMG2017] *diagramy przypadków użycia* zapewniają środki do modelowania systemu i jego kontekstu w zakresie realizowanych przypadków użycia systemu i aktorów w kontekście systemu, którzy wchodzi w interakcje z systemem poprzez te przypadki użycia.

Przypadki użycia modelują dynamiczną interakcję pomiędzy aktorem w kontekście systemu a systemem z perspektywy aktora. Przypadki użycia są w większości pisane przy użyciu szablonów w języku naturalnym (3.3) lub przy użyciu diagramów aktywności UML (3.4.4).

3.4.3 Modelowanie struktury i danych (L3)

Modele, które skupiają się na aspektach struktury i danych, określają wymagania dotyczące statycznych właściwości strukturalnych systemu lub domeny.

Statyczne modele domeny określają obiekty (biznesowe) i ich relacje w danej dziedzinie. Mogą być one wyrażone za pomocą diagramów klas UML [OMG2017].

Modele klas określają zasadniczo klasy systemu oraz ich atrybuty i relacje. Klasy reprezentują materialne i niematerialne encje występujące w świecie rzeczywistym, o których system musi posiadać dane, aby móc realizować swoje zadania. Diagramy klas UML są zazwyczaj używane jako język modelowania dla modeli klas.

3.4.4 Modelowanie funkcji i przepływu (L3)

Modele koncentrujące się na funkcji i przepływie określają wymagania dotyczące kolejności działań wymaganych do uzyskania żądanych wyników z danych wejściowych lub określają działania wymagane do realizacji procesu (biznesowego), włączając w to przepływ sterowania i danych pomiędzy działaniami oraz to, kto jest odpowiedzialny za jakie działania.

Modele aktywności są wykorzystywane do określania funkcji systemu. W języku modelowania UML [OMG2017] *diagramy aktywności* są używane do wyrażania modeli aktywności. Zawierają one elementy do modelowania aktywności i przepływu sterowania pomiędzy aktywnościami. Diagramy aktywności mogą również określać, kto jest odpowiedzialny za dane działania. Zaawansowane elementy modelowania (nieobjęte poziomem podstawowym CPRE) dostarczają środków do modelowania przepływu danych.

Modele procesu są wykorzystywane do opisu procesów biznesowych lub procesów technicznych. Można je wyrazić za pomocą diagramów aktywności UML lub za pomocą konkretnych języków modelowania procesów takich jak BPMN [OMG2013]. Na poziomie podstawowym CPRE do modelowania procesu używamy tylko diagramów aktywności UML.

3.4.5 Modelowanie stanu i zachowania (L2)

Modele, które skupiają się na stanie i zachowaniu, określają wymagania dotyczące zachowania systemu lub komponentu domeny z punktu widzenia zależnych od stanu reakcji na zdarzenia lub dynamikę interakcji komponentów.

Maszyny stanów modelują zdarzenia, które wyzwalają przejście z jednego stanu do drugiego oraz czynności, które muszą zostać wykonane w momencie przejścia w dany stan. *Schematy stanów* [Hare1988] to maszyny stanów ze zdekomponowanymi hierarchicznie i/lub ortogonalnie stanami. Maszyny stanów, w tym schematy stanów, mogą być wyrażone w języku modelowania UML [OMG2017] za pomocą *diagramów maszyny stanów* (zwanym również *diagramami stanów*).

3.4.6 Inne typy modeli w inżynierii wymagań (L1)

Na poziomie podstawowym CPRE zrozumienie i zastosowanie modeli jest ograniczone do wybranych, ważnych typów modeli. Istnieją inne typy modeli, które są wykorzystywane w inżynierii wymagań. Na poziomie podstawowym CPRE wystarczy je znać i wiedzieć, do czego one służą.

Modele celów reprezentują zbiór celów, celów podrzędnych i relacji między nimi. Modele celów mogą również obejmować zadania i zasoby potrzebne do osiągnięcia celu, aktorów, którzy chcą osiągnąć cel, oraz przeszkody, które utrudniają osiągnięcie celu.

W SysML [OMG2019] *diagramy definiowania bloków* (*block definition diagrams*) mogą być przystosowane do wyrażania diagramów kontekstowych przy użyciu stereotypowanych bloków dla systemu i aktorów. Diagramy definiowania bloków mogą być również wykorzystywane do modelowania struktury systemu w kategoriach jednostek koncepcyjnych systemu i relacji między nimi.

Modele historii domeny mogą być używane do modelowania funkcji i przepływu poprzez określenie wizualnych opowieści o tym, jak aktorzy wchodzi w interakcje z urządzeniami, artefaktami i innymi elementami (przedmiotami) w domenie, zwykle przy użyciu specyficznych dla danej domeny symboli [HoSch2020]. Są one środkiem do zrozumienia domeny aplikacji, w której będzie działać system.

Modele interakcji modelują dynamiczne interakcje między obiektami lub aktorami. *Diagramy sekwencji* UML są popularnym sposobem określania interakcji między obiektami.

3.5 Słowniki (L2)

W każdym przedsięwzięciu IW, które angażuje więcej niż jedną osobę, istnieje ryzyko braku wspólnego rozumienia terminologii — to znaczy, że niektóre osoby interpretują te same terminy na różne sposoby. Aby zminimalizować ten problem, w słowniku zapisuje się wspólne definicje istotnych terminów.

Słownik jest centralnym zbiorem definicji dla: terminów specyficznych dla kontekstu, terminów powszechnie używanych o specjalnym znaczeniu w danym kontekście, skrótów i akronimów. Synonimy (różne terminy oznaczające to samo) powinny być odpowiednio oznaczone. Należy unikać stosowania homonimów (używania tego samego terminu dla różnych obiektów – things) lub oznaczać je odpowiednio.

W przypadku słowników obowiązują następujące zasady:

- Zarządzaj słownikiem centralnie.
- Utrzymuj słownik przez cały czas rozwoju systemu.
- Wyznacz osobę lub małą grupę, która jest odpowiedzialna za słownik.
- Stosuj jednolity styl i strukturę w słowniku.
- Zaangażuj interesariuszy i dąż do porozumienia w sprawie terminologii.
- Udostępnij słownik wszystkim zainteresowanym.
- Narzuć obowiązek korzystania ze słownika.
- Sprawdzaj artefakty pod kątem prawidłowego korzystania ze słownika.

3.6 Dokumenty wymagań i struktura dokumentacji (L2)

Dokumenty specyfikacji wymagań (3.1.1) składają się z kilku artefaktów IW. Dlatego ważne jest, aby uporządkować takie dokumenty w dobrze zdefiniowanej strukturze, by stworzyć spójny i dający się utrzymać zbiór wymagań. Oprócz wymagań dokument wymagań może

również zawierać dodatkowe informacje i wyjaśnienia — na przykład słownik, warunki akceptacji, informacje o projekcie lub cechy implementacji technicznej.

Wymagania mogą być również zorganizowane w struktury dokumentacji inne niż klasyczne dokumenty.

Często używanymi dokumentami są:

- specyfikacje wymagań interesariuszy
- specyfikacje wymagań użytkownika (podzbiór specyfikacji wymagań interesariuszy obejmujący wyłącznie wymagania użytkowników)
- specyfikacje wymagań systemowych
- specyfikacje wymagań biznesowych
- dokument wizji.

Często stosowanymi, alternatywnymi strukturami dokumentacji, są:

- rejestr produktu (product backlog)
- rejestr sprintu (sprint backlog)
- mapa historyjek (story map).

Wybór struktury dokumentacji i wewnętrzna organizacja wybranej struktury dokumentu zależą od:

- wybranego procesu wytwarzania (5)
- typu domeny i typu procesu wytwarzania
- umowy (klient może zalecić korzystanie z danej struktury dokumentacji)
- rozmiaru dokumentu.

Szablony dokumentów mogą pomóc w uporządkowaniu specyfikacji wymagań. Szablony są dostępne w literaturze [Vole2020], [RoRo2012] oraz w standardach [ISO29148]. Można również wykorzystać szablony z poprzednich, podobnych projektów lub mogą być one narzucone przez klienta. Organizacja może również zdecydować się na stworzenie szablonu jako wewnętrznego standardu.

3.7 Prototypy w inżynierii wymagań (L1)

W IW *prototypy* są sposobem na określenie wymagań na konkretnym przykładzie oraz na walidację wymagań. W szczególności prototypy mogą być wykorzystywane, jeśli interesariusze nie chcą pisać i przeglądać artefaktów opartych na języku naturalnym, szablonach lub modelach.

Prototypy eksploracyjne (exploratory prototypes) [LiSZ1994] są wykorzystywane do osiągnięcia wspólnego zrozumienia, doprecyzowywania wymagań lub walidacji wymagań na różnych poziomach odwzorowania. Zazwyczaj są one niszczone po wykorzystaniu.

- *Szkielety (wireframes)* to prototypy o niskim poziomie odwzorowania, zbudowane z prostych materiałów lub przy użyciu narzędzi do szkicowania, które służą przede wszystkim do omawiania i walidacji pomysłów projektowych i koncepcji interfejsu użytkownika.

- *Makiety (mock-ups)* są prototypami o średnim poziomie odwzorowania. Przy określaniu systemów cyfrowych wykorzystuje się rzeczywiste ekrany i przepływy, ale bez rzeczywistej funkcjonalności. Służą one przede wszystkim do określania i walidacji interfejsów użytkownika.
- *Prototypy natywne (native prototypes)* są prototypami o wysokim poziomie odwzorowania, które implementują krytyczne części systemu w stopniu umożliwiającym interesariuszom sprawdzenie, czy prototypowana część systemu będzie działać i zachowywać się zgodnie z oczekiwaniami.

W zależności od poziomu odwzorowania, prototypy eksploracyjne mogą być drogimi artefaktami, więc zawsze należy mieć na uwadze uzgodnienie kompromisu pomiędzy kosztem a planowaną do uzyskania wartością.

Prototypy ewolucyjne (evolutionary prototypes) [LiSZ1994] to systemy pilotażowe. Najczęściej stanowią rdzeń systemu, który ma być opracowany. Ostateczny system ewoluuje poprzez stopniowe rozszerzanie i udoskonalanie systemu pilotażowego w kilku iteracjach.

3.8 Kryteria jakości artefaktów i wymagań (L1)

Wymagania muszą spełniać określone kryteria jakościowe, aby mogły być uznane za poprawnie zdefiniowane. W nowoczesnej IW, w podejściach zorientowanych na wartość (Zasada 1 w 2), stopień spełnienia kryterium jakości powinien odpowiadać wartości wytworzonej przez to wymaganie. Oznacza to, że wymagania nie muszą całkowicie spełniać wszystkich kryteriów jakościowych — jednakże im wyższa jest wartość wymagania, tym bardziej istotne są kryteria jakościowe, aby zmniejszyć ryzyko niepowodzenia.

Adekwatność i zrozumiałość są najważniejszymi kryteriami jakości dla pojedynczych wymagań. Bez nich wymaganie jest bezużyteczne lub nawet szkodliwe, niezależnie od spełnienia wszystkich innych kryteriów.

Kryteria jakości dla *pojedynczych wymagań* to:

- adekwatne (opisuje prawdziwe i uzgodnione potrzeby interesariuszy)
- niezbędne
- jednoznaczne
- kompletne (niezależne)
- zrozumiałe
- weryfikowalne.

Jak opisano powyżej w 3.1.1, wymagania są zazwyczaj zapisywane w różnych artefaktach, które obejmują pojedyncze wymagania lub wiele wymagań. Powyższe kryteria jakości należy wykorzystywać w celu stworzenia pojedynczych wymagań wysokiej jakości w artefakcie. Dla artefaktów, które obejmują więcej niż jedno wymaganie, należy dodatkowo uwzględnić poniższe kryteria jakości.

Kryteria jakości dla artefaktów obejmujących *wiele wymagań*:

- spójne
- nie są nadmiarowe (brak redundancji)
- kompletne (nie pominięto żadnych znanych i istotnych wymagań)
- modyfikowalne
- możliwe do śledzenia (traceable)
- zgodne z daną specyfikacją lub standardem.

4 Praktyki w zakresie opracowywania wymagań

(L3)

Cel: Zrozumienie stosowania praktyk dotyczących identyfikacji źródeł wymagań, pozyskiwania wymagań, identyfikowania i rozwiązywania konfliktów i walidacji wymagań.

Czas trwania: 4 godziny 30 minut

Terminy: Źródło wymagań, granica systemu, kontekst systemu, pozyskiwanie wymagań, walidacja wymagań, interesariusz, Model Kano, rozwiązywanie konfliktu

Cele nauczania

- EO 4.1.1 Umiejętność wyznaczenia granic systemu w celu skupienia się na istotnych wymaganiach (L3)
- EO 4.1.2 Znajomość istotnych źródeł wymagań dla tworzonego systemu (L1)
- EO 4.1.3 Umiejętność zidentyfikowania interesariuszy i sporządzenia listy interesariuszy (L3)
- EO 4.1.4 Zrozumienie korzyści związanych z zarządzaniem interesariuszami (L2)
- EO 4.2.1 Zrozumienie roli Modelu Kano w pozyskaniu właściwych wymagań (L2)
- EO 4.2.2 Zrozumienie różnicy pomiędzy technikami gromadzenia a technikami projektowania i generowania pomysłów (L2)
- EO 4.2.3 Zrozumienie, w jaki sposób wybrać odpowiednią technikę pozyskiwania wymagań w danej sytuacji (L2)
- EO 4.3.1 Znajomość różnych rodzajów konfliktów (L1)
- EO 4.3.2 Zrozumienie, jakie działania są niezbędne do rozwiązywania konfliktów (L2)
- EO 4.3.3 Zrozumienie, w jaki sposób zastosować odpowiednie techniki rozwiązywania konfliktów (L2)
- EO 4.4.1 Zrozumienie, dlaczego dokumenty wymagań muszą być walidowane (L2)
- EO 4.4.2 Znajomość czterech ważnych aspektów walidacji wymagań (L1)
- EO 4.4.3 Zrozumienie, w jaki sposób zastosować odpowiednie techniki walidacji wymagań (L2).

4.1 Źródła wymagań (L3)

Jakość i kompletność wymagań zależy w dużej mierze od zdefiniowania odpowiednich źródeł wymagań. Pominięcie istotnego źródła doprowadzi do niekompletnych wymagań lub niepełnego zdefiniowania wymagań. Identyfikacja źródeł wymagań jest iteracyjnym i rekurencyjnym (powtarzalnym) procesem, wymagającym ciągłej aktualizacji.

Wspólne zrozumienie (Zasada 3 w 2) kontekstu systemu, który ma zostać wytworzony, jest warunkiem wstępnym określenia odpowiednich źródeł wymagań. Obszar pomiędzy granicą systemu a granicą kontekstu nazywany jest kontekstem (systemu) (Zasada 4 w 2). Kontekst (systemu) jest potrzebny do zrozumienia natury wymagań, które mają być wytworzone i tym samym zidentyfikowania pierwotnych źródeł wymagań.

Źródła wymagań dzielą się na trzy rodzaje:

- interesariusze
- dokumenty
- systemy.

Interesariusze systemu (patrz definicja [Glin2020], patrz także: Zasada 2 w 2) są głównym źródłem wymagań. Typowe role interesariuszy obejmują [BiSp2003]:

- użytkowników (zwanym również użytkownikami końcowymi)
- sponsorów
- kierowników
- deweloperów
- właściwe jednostki administracyjne
- klientów.

Ponadto osoby lub organizacje, na które system wywiera *wpływ*, powinny być traktowane jako (pośredni) interesariusze.

Systematyczna identyfikacja interesariuszy powinna mieć miejsce na początku projektu, a rezultatami należy zarządzać przez cały okres wytwarzania jak ciągłą aktywnością. Obejmuje to identyfikację zarówno ról interesariuszy, jak i osób pełniących te role.

Dla wszystkich systemów posiadających interfejs użytkownika, *użytkownicy końcowi* systemu stanowią grupę interesariuszy będącą w obszarze szczególnego zainteresowania inżyniera wymagań. Użytkownicy końcowi powinni być łączeni w grupy (na przykład według podobnych ról, zadań lub odpowiedzialności).

Gdy użytkownicy końcowi mogą być zidentyfikowani indywidualnie, należy wybrać przedstawicieli każdej z grup. W innym przypadku można zdefiniować osoby dla reprezentacji odpowiednich grup użytkowników końcowych [Coop2004].

Potencjalnymi źródłami, pomocnymi w identyfikacji istotnych interesariuszy i ról interesariuszy, są:

- listy typowych grup interesariuszy i ról
- struktury organizacyjne
- dokumentacja procesu biznesowego
- raporty rynkowe
- początkowi interesariusze w celu zidentyfikowania *dodatkowych* interesariuszy.

Interesariusze powinni być udokumentowani za pomocą aktualnej listy interesariuszy, zawierającej (co najmniej) następujące informacje:

- nazwisko
- funkcja (rola)
- dodatkowe dane personalne i kontaktowe
- dostępność czasowa i przestrzenna w trakcie postępu projektu
- znaczenie
- obszar i zakres wiedzy specjalistycznej
- cele i interesy w odniesieniu do projektu.

Problemy z interesariuszami mogą się pojawić, jeśli ich prawa i obowiązki nie są jasne lub jeśli ich potrzeby nie są wystarczająco uwzględnione. Zarządzanie relacjami z interesariuszami [Bour2009] jest skutecznym sposobem przeciwdziałania problemom związanym z interesariuszami.

W większości kontekstów systemu dostępnych jest więcej źródeł. Należy je również brać pod uwagę przy tworzeniu nowego, udanego systemu, ponieważ większość interesariuszy nie mówi o oczywistych, ich „podświadomych” wymaganiach (4.2).

Dodatkowe źródła wymagań obejmują:

- istniejące i dotychczasowe systemy
- dokumenty procesów
- dokumenty prawne lub regulacyjne
- regulacje specyficzne dla firmy
- (marketingowe) informacje o potencjalnych, przyszłych użytkownikach.

Kolejne źródła wymagań mogą być odkryte poprzez przyjrzenie się podobnym sytuacjom w zupełnie innych dziedzinach.

4.2 Pozyskiwanie wymagań (L2)

Zadaniem inżyniera wymagań w ramach pozyskiwania wymagań jest zrozumienie oczekiwań i potrzeb interesariuszy, przy jednoczesnym zapewnieniu, że wymagania pochodzące ze wszystkich istotnych źródeł wymagań zostały pozyskane poprzez zastosowanie odpowiednich technik ich pozyskiwania. Głównym punktem w pozyskiwaniu wymagań jest przekształcenie domniemych żądań, życzeń i oczekiwań w konkretne wymagania.

Aby pozyskać wymagania, niezbędna jest znajomość natury i znaczenia każdego wymagania. Te mogą się zmieniać w zależności od projektu, a także z czasem. Model Kano [KAA1984] dzieli wymagania na trzy istotne kategorie:

- czynniki entuzjazmu (synonimy: wzbudzające zachwyty, wymagania nieświadomione)
- czynniki wydajności (satisfiers) (synonimy: wzbudzające zadowolenie, wymagania uświadomione)
- czynniki podstawowe (dissatisfiers) (synonimy: wzbudzające niezadowolenie, wymagania podświadome).

Istnieje wiele różnych technik pozyskiwania wymagań należących do tych kategorii. Wśród nich wyróżniamy:

- techniki gromadzenia
- techniki projektowania i generowania pomysłów.

Techniki gromadzenia są ustanowionymi technikami pozyskiwania wymagań [BaCC2015], pomagającymi pozyskać czynniki wydajności (satisfiers) oraz czynniki podstawowe (dissatisfiers) poprzez badanie różnych źródeł.

Można wyróżnić cztery główne kategorie:

- techniki zadawania pytań
- techniki współpracy
- techniki obserwacji
- techniki oparte na artefaktach.

Techniki projektowania i generowania pomysłów mają na celu pobudzenie kreatywności podczas pozyskiwania wymagań. Ich celem jest tworzenie pomysłów na rozwiązanie problemu oraz eksplorowanie idei projektowych [Kuma2013]. Może to prowadzić do nowych lub innowacyjnych wymagań, które często wzbudzają zachwyty. Popularne przykłady takich technik to burza mózgów [Osbo1979], analogie, prototypowanie (np. makiety), scenariusze i tablice historyjek (storyboards).

Szerszym pojęciem związanym z projektowaniem i generowaniem pomysłów jest *myślenie projektowe (design thinking)*. Istnieją różne podejścia, takie jak *d.school* [Sdsc2012] i *Designing for Growth* [LiOg2011], oferujące szeroki zestaw technik, które mogą być wykorzystane do pozyskiwania wymagań.

Techniki pozyskiwania wymagań powinny być w stanie wykryć wszystkie rodzaje wymagań – zarówno funkcjonalne, jak i jakościowe, a także ograniczenia. W praktyce wymaganiom jakościowym i ograniczeniom często poświęca się mniej uwagi.

Aby pozyskać *wymagania jakościowe* jako listę kontrolną powinno się użyć modelu jakości takiego jak np. norma ISO 25010 [ISO25010]. Ten model może być również przydatny do ilościowego opisu (kwantyfikacji) wymagań.

Ograniczenia mogą zostać odkryte poprzez rozważenie możliwych restrykcji dotyczących przestrzeni potencjalnego rozwiązania – na przykład zagadnień technicznych, prawnych, organizacyjnych, kulturowych lub środowiskowych.

Wybór właściwych technik pozyskiwania wymagań jest krytyczną, kluczową kompetencją inżyniera wymagań, która zależy od wielu różnych czynników takich jak:

- rodzaj systemu
- model cyklu wytwarzania oprogramowania
- zaangażowane osoby
- struktura organizacji.

Najlepsze wyniki osiąga się zazwyczaj przy zastosowaniu kombinacji różnych technik pozyskiwania wymagań. [CaDJ2014] przedstawia systematyczne podejście do wyboru technik.

4.3 Rozwiązywanie konfliktów dotyczących wymagań (L2)

Same techniki pozyskiwania wymagań nie zapewniają spójności, kompletności, zgodności z daną specyfikacją lub standardem itp. powstałego zbioru wymagań (3.8). Ostatecznie wszyscy interesariusze muszą zrozumieć i uzgodnić wszystkie istotne dla nich wymagania. Jeśli niektórzy interesariusze nie zgadzają się, sytuację tę należy uznać za konflikt, który powinien zostać odpowiednio rozwiązany. Na podstawie rodzaju konfliktu i informacji

wynikających z kontekstu powinny być wybierane odpowiednie techniki rozwiązywania konfliktów. Wymaga to dogłębnego zrozumienia charakteru konfliktu wymagań oraz postawy zaangażowanych interesariuszy.

Zadania w zakresie identyfikacji i rozwiązywania konfliktów to:

- rozpoznanie konfliktu
- analiza konfliktu
- rozwiązanie konfliktu
- dokumentacja rozwiązania konfliktu (podjęte decyzje).

Przydatne jest rozróżnienie pomiędzy różnymi rodzajami konfliktów [Moor2014]. Rodzaje konfliktów, na które inżynier wymagań często musi zwrócić uwagę, to:

- konflikt merytoryczny
- konflikt danych
- konflikt interesu
- konflikt wartości
- konflikt relacji
- konflikt strukturalny.

Aby skutecznie rozwiązywać konflikty, można stosować popularne techniki:

- zgoda
- kompromis
- głosowanie
- uchylene
- definicja wariantów.

Ponadto istnieje kilka technik pomocniczych, na przykład:

- rozważenie wszystkich faktów
- rozważenie plusów i minusów różnych opcji
- macierz decyzyjna.

4.4 Walidacja wymagań (L2)

Walidacja wymagań jest ważnym krokiem w kierunku udanego systemu (Zasada 6 w 2).

Zapewnienie jakości wymagań od początku projektu ograniczy marnotrawstwo wysiłku na dalszych etapach projektu. Walidacja wymagań oznacza sprawdzenie pod względem jakości zarówno artefaktów, jak i poszczególnych wymagań w nich zawartych (szczegóły w podrozdziale 3.8).

Ważnymi aspektami, które należy wziąć pod uwagę podczas walidacji wymagań, są:

- zaangażowanie właściwych interesariuszy
- oddzielenie diagnozy od korekty defektów
- walidacja z różnych perspektyw
- powtórna walidacja.

Istnieje kilka technik walidacji (np. [GiGr1993], [OleA2018]). Techniki walidacji są często klasyfikowane w następujący sposób:

- *techniki przeglądu (review)*, w tym:
 - przeglądnienia
 - inspekcje
- *techniki eksploracyjne*, np.:
 - prototypowanie
 - testowanie alfa i testowanie beta
 - testowanie A/B [KoTh2017]
 - budowanie minimalnego, użytecznego produktu (MVP – Minimum Viable Product)
- *próbne opracowanie rozwiązania*.

Techniki te różnią się pod względem stopnia sformalizowania i wysiłku. Wybór techniki zależy od czynników takich jak: model cyklu wytwarzania oprogramowania, dojrzałość procesu wytwarzania, złożoność i poziom ryzyka systemu, dowolne wymagania prawne i regulacyjne i/lub potrzeby związane ze ścieżką audytu.

5 Proces i struktura pracy [L3]

Cel: Wyjaśnienie pojęć procesu IW i zastosowania odpowiednich konfiguracji procesu

Czas trwania: 1 godzina 15 minut

Terminy: Proces, proces IW

Cele nauczania

- EO 5.1.1 Znajomość ważnych czynników, które wpływają na proces IW (L1)
- EO 5.1.2 Zrozumienie, dlaczego i w jaki sposób te czynniki wpływają na proces IW (L2)
- EO 5.2.1 Umiejętność określenia aspektów, które należy uwzględnić przy konfiguracji procesu IW (L2)
- EO 5.3.1 Znajomość typowych konfiguracji procesu IW (L1)
- EO 5.3.2 Umiejętność określenia kroków konfigurowania procesu IW (L2)
- EO 5.3.3 Umiejętność wyboru i zastosowania odpowiedniej konfiguracji procesu IW dla prostego systemu i uwarunkowań projektowych (L3)

Aby ukształtować i usystematyzować pracę w ramach IW wykonywanej w danym kontekście, potrzebny jest odpowiedni proces. Ponieważ nie istnieje jeden uniwersalny proces IW (1.4), należy skonfigurować odpowiedni, dopasowany do danego kontekstu i systemu, proces IW.

Proces IW definiuje przepływ informacji i model komunikacji pomiędzy różnymi uczestnikami (np. klientami, użytkownikami, inżynierami wymagań, programistami, testerami), a także definiuje artefakty, które mają być używane lub wytworzone. W ten sposób proces IW zapewnia ramy (wzorce projektowe – frameworks) dla pozyskiwania, dokumentowania, walidowania i zarządzania wymaganiami.

5.1 Czynniki wpływające [L2]

Wiele czynników wpływa na konfigurację procesu IW. Do głównych czynników zaliczamy:

- ogólne dopasowanie procesu: proces IW musi być dopasowany do ogólnego (całościowego) procesu wytwarzania systemu
- kontekst wytwarzania systemu
- możliwości i dostępność interesariuszy
- wspólne zrozumienie
- złożoność i krytyczność systemu, który ma zostać wytworzony
- ograniczenia
- dostępny czas i budżet
- zmienność wymagań
- doświadczenie inżynierów wymagań.

Analiza czynników wpływających dostarcza informacje użyteczne podczas konfiguracji procesu IW. Czynniki wpływające ograniczają liczbę możliwych konfiguracji tego procesu. Na przykład, gdy interesariusze są dostępni tylko na początku projektu, nie można wybrać procesu, który opiera się na stałej informacji zwrotnej od interesariuszy.

5.2 Aspekty procesu inżynierii wymagań (L2)

Istnieją trzy decydujące aspekty, które należy wziąć pod uwagę podczas konfigurowania procesu IW [Glin2019].

Aspekt czasu: liniowy versus iteracyjny

W procesie liniowym (linear) wymagania są określone z góry we wskazanej fazie procesu. W procesie iteracyjnym (iterative) wymagania są określane przyrostowo, począwszy od ogólnych celów i kilku wymagań początkowych poprzez dodawanie lub modyfikację wymagań w każdej iteracji.

Kryteria wyboru *liniowego* procesu IW:

- Proces wytwarzania systemu jest sterowany planem (plan-driven) i w większości przypadków liniowy.
- Interesariusze znają swoje wymagania i potrafią je z góry określić.
- Kompletna specyfikacja wymagań jest wymagana jako podstawa kontraktu w przypadku outsourcingu projektowania i wdrożenia systemu.
- Organy regulacyjne wymagają kompletnej, formalnie dopuszczonej specyfikacji wymagań na wczesnym etapie wytwarzania.

Kryteria wyboru *iteracyjnego* procesu IW:

- Proces wytwarzania systemu jest iteracyjny i zwinny.
- Wiele wymagań nie jest znanych z góry, natomiast pojawią się one i będą ewoluować w trakcie wytwarzania systemu.
- Interesariusze są na tyle dostępni, że można ustanowić krótkie pętle informacji zwrotnej jako środek zmniejszający ryzyko wytworzenia niewłaściwego systemu.
- Czas trwania (etapu) wytwarzania pozwala na więcej niż jedną lub dwie iteracje.
- Ważna jest możliwość łatwej zmiany wymagań.

Aspekt celu: proces nakazowy versus eksploracyjny

W nakazowym (prescriptive) procesie IW specyfikacja wymagań stanowi umowę: wszystkie wymagania są wiążące i muszą być wdrożone. W eksploracyjnym (explorative) procesie IW z góry znane są tylko cele, podczas gdy konkretne wymagania muszą być określone.

Kryteria wyboru *nakazowego* procesu IW:

- Klient wymaga ścisłej umowy dotyczącej wytworzenia systemu.
- Funkcjonalność i zakres mają pierwszeństwo przed kosztami i terminami.
- Wytworzenie określonego systemu może być przedmiotem przetargu lub outsourcingu.

Kryteria wyboru *eksploracyjnego* procesu IW:

- Interesariusze początkowo mają jedynie mgliste pojęcie o swoich wymaganiach.
- Interesariusze są mocno zaangażowani i stale udzielają informacji zwrotnych.
- Terminy i koszty mają pierwszeństwo przed funkcjonalnością i zakresem.

- Nie jest z góry jasne, które wymagania faktycznie powinny być wdrożone i w jakiej kolejności będą wdrożone.

Aspekt docelowego odbiorcy: zorientowanie na klienta versus zorientowanie na rynek

W procesie IW zorientowanym na klienta (customer-specific) system jest zamawiany przez klienta i wytwarzany przez dostawcę. W procesie IW zorientowanym na rynek (market-oriented) system jest wytwarzany jako produkt lub usługa dla rynku i jest ukierunkowany na konkretne segmenty użytkowników.

Kryteria wyboru procesu IW *zorientowanego na klienta*:

- System będzie używany głównie przez organizację, która zamówiła system i płaci za jego wytworzenie.
- Ważni interesariusze w większości są związani z organizacją klienta.
- Poszczególne osoby mogą być identyfikowane jako interesariusze.
- Klient chce otrzymać specyfikację wymagań, która może służyć jako umowa.

Kryteria wyboru procesu IW *zorientowanego na rynek*:

- Organizacja wytwarzająca (system) zamierza go sprzedawać w jakimś segmencie rynku jako produkt lub usługę.
- Potencjalni użytkownicy nie są indywidualnie identyfikowalni.
- Inżynierowie wymagań muszą tak opracować wymagania, aby odpowiadały one przewidywalnym potrzebom docelowych użytkowników.
- Właściciele produktów (Product Owners), ludzie odpowiedzialni za marketing, cyfrowi projektanci i architekci systemów są głównymi interesariuszami.

Wskazówki i zastrzeżenia

- Przedstawione powyżej kryteria to raczej *heurystyki* niż ustalone zasady. Na przykład outsourcing wytwarzania systemu realizowany jest najczęściej przy pomocy procesu IW o charakterze nakazowym, a nie eksploracyjnym, ponieważ umowa pomiędzy klientem a dostawcą jest zazwyczaj oparta na kompleksowej specyfikacji wymagań. Możliwe jest jednak również negocjowanie umowy outsourcingowej w oparciu o eksploracyjny proces IW.
- Procesy liniowe IW mają zastosowanie tylko wtedy, gdy istnieje skomplikowany proces zmiany wymagań.
- Procesy liniowe IW implikują długie pętle informacji zwrotnej: aby zminimalizować ryzyko wytworzenia niewłaściwego systemu, wymagania muszą być intensywnie walidowane.
- Podczas definiowania procesu IW proces *liniowy* i *nakazowy* często są wybierane razem.
- Procesy eksploracyjne IW są zazwyczaj również procesami iteracyjnymi (i na odwrót).
- W procesie zorientowanym na rynek informacja zwrotna od użytkowników jest jedynym środkiem potwierdzającym, czy produkt rzeczywiście zaspokoi potrzeby docelowego segmentu użytkowników.
- Aspekt orientacji na rynek nie łączy się dobrze z aspektami liniowym i nakazowym.

5.3 Konfigurowanie procesu inżynierii wymagań (L3)

W konkretnym kontekście wytwarzania systemu, osoby odpowiedzialne za IW muszą skonfigurować proces IW, który ma być zastosowany. W oparciu o analizę czynników wpływających (opisanych) w (5.1) można zastosować odpowiednią kombinację opisanych aspektów procesu 5.2 [Glin2019]. Poniżej opisano trzy typowe kombinacje.

Angażujący proces IW: iteracyjny, eksploracyjny i zorientowany na klienta

Główny przypadek zastosowania:	dostawca i klient ściśle ze sobą współpracują, interesariusze są silnie zaangażowani w procesy zarówno IW, jak i wytwarzania systemu.
Typowe artefakty:	rejestr produktu (product backlog) zawierający historyjki użytkowników (user stories) i/lub opisy zadań, prototypy.
Typowy przepływ informacji:	ciągła interakcja pomiędzy interesariuszami, właścicielami produktów (Product Owners), inżynierami wymagań a deweloperami; może obejmować informacje zwrotne od użytkowników.

Kontraktowy proces IW: typowo liniowy (czasami iteracyjny), nakazowy i zorientowany na klienta

Główny przypadek zastosowania:	specyfikacja wymagań stanowi podstawę umowy na wytworzenie systemu przez osoby niezaangażowane w opracowanie specyfikacji, z niewielką interakcją z interesariuszami po fazie (pozyskiwania) wymagań.
Typowe artefakty:	klasyczna specyfikacja wymagań systemowych składająca się z wymagań w języku naturalnym i modeli.
Typowy przepływ informacji:	głównie od interesariuszy do inżynierów wymagań.

Zorientowany na produkt proces IW: iteracyjny i eksploracyjny oraz zorientowany na rynek

Główny przypadek zastosowania:	organizacja specyfikuje i wytwarza oprogramowanie w celu sprzedaży lub dystrybucji jako produkt lub usługę.
Typowe artefakty:	rejestr produktu (product backlog), prototypy.
Typowy przepływ informacji:	interakcja pomiędzy właścicielem produktu (Product Owner), marketingiem, inżynierami wymagań, projektantami, deweloperami i (być może) szybkie informacje zwrotne od klientów/użytkowników.

Należy pamiętać, że mogą istnieć konteksty systemu i wytwarzania, którym nie odpowiada żadna z wyżej wymienionych konfiguracji. Na przykład ograniczenia regulacyjne mogą nakładać obowiązek stosowania procesu zgodnego z określonymi normami, takimi jak np. ISO/IEC/IEEE 29148 [ISO29148].

Podczas konfigurowania procesu IW zalecane jest zastosowanie pięcioetapowej procedury:

1. Analiza czynników wpływających (5.1).
2. Ocena kryteriów aspektu (5.2).
3. Konfiguracja procesu (5.3).
4. Określenie artefaktów (3).
5. Wybór odpowiedniej praktyki.

6 Praktyki w zakresie zarządzania

wymaganiami (L2)

Cel: Zrozumienie potrzeb i korzyści płynących z zarządzania wymaganiami

Czas trwania: 2 godziny

Terminy: Zarządzanie wymaganiami, zarządzanie zmianami, śledzenie powiązań pomiędzy wymaganiami (traceability), atrybuty wymagań, cykl życia wymagań, priorytetyzacja

Cele nauczania

- EO 6.1.1 Umiejętność określenia, na czym polega zarządzanie wymaganiami i dlaczego jest ono potrzebne (L1)
- EO 6.2.1 Umiejętność zdefiniowania przyczyn, z powodu których artefakty wymagają określenia ich statusu oraz modelu cyklu życia (L2)
- EO 6.3.1 Umiejętność wyjaśnienia koncepcji wersjonowania wymagań w danej sytuacji projektowej (L2)
- EO 6.4.1 Znajomość zastosowania konfiguracji wymagań i wersji podstawowych (baselines) (L1)
- EO 6.5.1 Znajomość zastosowania atrybutów wymagań (L1)
- EO 6.5.2 Umiejętność wyjaśnienia, jak wygląda odpowiedni zestaw atrybutów wymagań w danej sytuacji projektowej (L2)
- EO 6.5.3 Umiejętność określenia zastosowania widoków (views) wymagań i nazwania różnych widoków wymagań (L2)
- EO 6.6.1 Umiejętność określenia celu śledzenia powiązań pomiędzy wymaganiami (L1)
- EO 6.6.2 Umiejętność określenia (podsumowania) różnic pomiędzy domniemanym a wyraźnym śledzeniem powiązań pomiędzy wymaganiami (L1)
- EO 6.6.3 Umiejętność określenia sposobu udokumentowania wyraźnego śledzenia powiązań pomiędzy wymaganiami (L1)
- EO 6.7.1 Umiejętność określenia sposobów radzenia sobie ze zmianami wymagań w podejściach sterowanych planem (plan-driven) i zwinnych (Agile) (L1)
- EO 6.8.1 Znajomość przyczyny priorytetyzacji wymagań i istotnych kryteriów oceny (L1)
- EO 6.8.2 Znajomość kroków priorytetyzacji wymagań (L1)
- EO 6.8.3 Umiejętność określenia różnych kategorii technik priorytetyzacji wymagań (L1).

6.1 Czym jest zarządzanie wymaganiami? (L1)

Zarządzanie wymaganiami jest procesem zarządzania istniejącymi wymaganiami zapisanymi w różnych artefaktach. W szczególności obejmuje ono: gromadzenie, zmianę i śledzenie powiązań pomiędzy wymaganiami [Glin2020]. Zarządzanie wymaganiami może odbywać się na różne sposoby i na różnych poziomach, w zależności od wybranego procesu wytwarzania (development process) i kontekstu — patrz [Leff2011], [Rupp2014], [WiBe2013]. Niezależnie od okoliczności, zadaniem zarządzania wymaganiami jest utrzymywanie wymagań w taki sposób, aby wszyscy biorący udział w projekcie mogli działać skutecznie i efektywnie.

6.2 Zarządzanie cyklem życia wymagań (L2)

Zarządzanie cyklem życia polega na śledzeniu wszystkich artefaktów w odniesieniu do ich aktualnego stanu w ramach ich cyklu życia. Każde udokumentowane wymaganie i każdy artefakt ma swój własny cykl życia: jest tworzony, następnie oceniany i udoskonalany, zanim zostanie poddany przeglądowi, przerobiony, scalony, uzgodniony itd. Aby umożliwić określenie, który artefakt jest w jakim stanie, konieczne jest zdefiniowanie modelu cyklu życia określającego każdy możliwy stan i warunki przejścia między stanami. Faktyczny stan artefaktu powinien być zawsze znany, (zazwyczaj) łącznie z historią jego przejść.

6.3 Kontrola wersji (L2)

Kontrola wersji wymagań odnosi się do procesu śledzenia zmian wszystkich artefaktów podczas ich rozwijania. Wszelkie zmiany w artefakcie powinny być odzwierciedlone w jego nowej wersji. Wersjonowanie pozwala na prześledzenie historii zmian artefaktu od momentu jego powstania i przywrócenie go do dowolnej, wcześniejszej wersji. W tym celu kontrola wersji wymaga zastosowania trzech środków:

- numeru wersji umożliwiającego jednoznaczny identyfikację wersji artefaktu
- historii tego, co zostało zmienione.
- koncepcji przechowywania artefaktów.

Wersjonowanie należy rozważyć biorąc pod uwagę wszystkie artefakty [WiBe2013]. Numer wersji zazwyczaj składa się z co najmniej dwóch części: wersji i przyrostu.

6.4 Konfiguracje i wersje podstawowe (baselines) (L1)

Konfiguracja wymagań jest spójnym zestawem artefaktów zawierających wymagania. Każda konfiguracja jest zdefiniowana w konkretnym celu i zawiera co najwyżej jedną wersję każdego artefaktu [Glin2020]. Celem konfiguracji wymagań może być na przykład przegląd wybranego zestawu artefaktów lub ułatwienie oszacowania wysiłku związanego z wytwarzaniem.

Wersją podstawową (punktem odniesienia – *baseline*) jest stabilna, kontrolowana pod względem zmian (change-controlled) *konfiguracja* artefaktów, używana do planowania wydania (release) lub innych etapów projektu (milestones) [Glin2020].

Konfiguracje mają następujące właściwości:

1. logiczne powiązanie
2. spójność
3. unikalność
4. niezmienność
5. możliwość powrotu do poprzedniej wersji.

6.5 Atrybuty i widoki (L2)

Atrybuty są wymagane do dokumentowania istotnych metadanych dotyczących artefaktów i są zazwyczaj używane do udzielania odpowiedzi na wiele ważnych pytań pojawiających się podczas cyklu życia projektu lub produktu.

Celem wykorzystania atrybutów do określania wymagań jest umożliwienie członkom zespołu i innym interesariuszom uzyskania informacji na temat wymagań, które mogą być im potrzebne w dowolnym momencie trwania projektu.

Zdefiniowanie odpowiedniego zestawu atrybutów zależy od potrzeb informacyjnych różnych interesariuszy w projekcie. Istniejące standardy, na przykład [ISO29148], prezentują przegląd najbardziej istotnych atrybutów.

Widoki (*views*) są fragmentami całego zestawu wymagań, który zawiera tylko te treści, które są aktualnie interesujące. Z technicznego punktu widzenia, widok jest kombinacją ustawień filtrów i sortowania, którą można udostępnić innym lub ponownie wykorzystać poprzez zapisanie wybranej kombinacji.

Rozróżniamy trzy rodzaje widoków:

- *widoki selektywne (selective)*
- *widoki projekcyjne (projective)*
- *widoki sumaryczne (aggregating).*

W większości przypadków widoki wymagań są kombinacją widoku selektywnego, projekcyjnego i sumarycznego używaną do tworzenia raportów.

6.6 Śledzenie powiązań pomiędzy wymaganiami (traceability) (L1)

Śledzenie powiązań pomiędzy wymaganiami (*traceability*) [GoFi1994] to zdolność do prześledzenia *pochodzenia* danego wymagania (tj. interesariuszy, dokumentów, uzasadnień itp.) i *powiązania go* (jako wkład) z *innymi artefaktami* (np. *przypadkami testowymi*), które pozwala również na powiązanie wymagania z *innymi wymaganiami*, od których ono zależy.

Śledzenie powiązań pomiędzy wymaganiami jest warunkiem wstępnym zarządzania wymaganiami i często jest wyraźnie wymagane przez normy, prawo i przepisy. Wdrożenie śledzenia powiązań (identyfikowalności) oznacza zasadniczo utrzymanie zależności między różnymi artefaktami (3.1) na różnych poziomach abstrakcji (3.1.2), poziomach szczegółowości (3.1.3) oraz ze wszystkimi istotnymi poprzednikami i następcami dla potrzeb analizy, zgodności i informacji.

Śledzenie powiązań pomiędzy wymaganiami może być udokumentowane w sposób *domniemany* poprzez ustrukturyzowanie i standaryzację artefaktów lub w sposób *wyraźny (jawny)* poprzez wzajemne powiązanie artefaktów, w oparciu o ich unikalne identyfikatory występujące w różnych formach [HuJD2011]. Powszechne formy reprezentacji relacji to hiperłącza, odwołania, matryce, tabele lub wykresy.

6.7 Obsługa zmiany (L1)

Wymagania nie są statyczne. Zmiany wymagań wynikają z wielu różnych przyczyn i muszą być odpowiednio obsługiwane (Zasada 7 w 2), na przykład poprzez stworzenie formalnego wniosku o zmianę (*change request*) lub dodanie nowej pozycji do rejestru produktu (*product backlog*).

Podejmowanie decyzji, planowanie i kontrolowanie wdrażania zmiany zależy od podejścia do wytwarzania i momentu, w którym zmiana się pojawia.

W podejściu *liniowym* decyzja o zmianie jest często podejmowana przez Radę Kontroli Zmian (Change Control Board) – na poziomie projektowym lub przez Radę Doradztwa Zmian (Change Advisory Board) – na poziomie operacyjnym. W bardziej *iteracyjnym* podejściu, właściciel produktu (Product Owner) uwzględnia zmianę w rejestrze produktu (*product backlog*) i nadaje odpowiedni priorytet nowej pozycji.

6.8 Priorytetyzacja (L1)

Nie wszystkie wymagania są jednakowo ważne [Davi2005]. Ocena i priorytetyzacja są wykorzystywane do określenia najistotniejszych wymagań, które mają być ujęte w następnym wydaniu lub przyroście (increment).

Ocena wymagań jest podstawą do ich priorytetyzacji, często określanej przy użyciu wielu kryteriów oceny takich jak wartość biznesowa, pilność, wysiłek, zależności i inne.

Priorytet wymagania opisuje znaczenie pojedynczego wymagania w porównaniu z innymi wymaganiami, według określonych kryteriów [Glin2020]. Sama *priorytetyzacja* jest przeprowadzana na podstawie jednego kryterium lub wielu kryteriów – zależy to głównie od wybranej techniki priorytetyzacji.

Wyróżniamy następujące kroki priorytetyzacji:

- zdefiniowanie głównych celów i ograniczeń dotyczących ustalania priorytetów
- określenie pożądaných kryteriów oceny
- określenie interesariuszy, którzy mają być zaangażowani
- określenie wymagań, którym mają być nadane priorytety
- wybór techniki priorytetyzacji
- przeprowadzanie priorytetyzacji.

Techniki priorytetyzacji dzielą się na:

- techniki priorytetyzacji *ad-hoc*
- *analityczne* techniki priorytetyzacji.

7 Narzędzia wspierające [L2]

Cel: Omówienie roli narzędzi w IW i aspektów ich wdrażania

Czas trwania: 30 minut

Terminy: Narzędzie, narzędzie IW

Cele nauczania

EO 7.1.1 Znajomość różnych rodzajów narzędzi IW (L1)

EO 7.2.1 Umiejętność określenia zagadnień, które należy uwzględnić przy wprowadzaniu narzędzi IW (L2)

7.1 Narzędzia w inżynierii wymagań [L1]

Proces IW może być wspierany przez narzędzia, które wspierają przypisane (temu procesowi) zadania i czynności. Ponieważ proces IW jest dość niepowtarzalny (5), istniejące narzędzia często skupiają się tylko na pewnych aspektach w IW i rzadko wspierają wszystkie czynności. Przed wyborem narzędzia inżynierowie wymagań powinni zdecydować, które zadania i czynności w procesie IW powinny być wspierane i w jaki sposób. Wyróżniamy narzędzia, które wspierają:

- Zarządzanie wymaganiami:
 - definiowanie i przechowywanie atrybutów wymagań
 - priorytetyzację wymagań
 - zarządzanie wersjami i konfiguracjami
 - śledzenie powiązań pomiędzy wymaganiami (traceability)
- zarządzanie zmianami w wymaganiach.
 - Zarządzanie procesem IW:
 - pomiary i raportowanie procesu
 - pomiary i raportowanie jakości produktu
- zarządzanie przepływem pracy IW.
 - Dokumentowanie wiedzy na temat wymagań:
 - udostępnianie wymagań
 - kreowanie wspólnego rozumienia wymagań.
- modelowanie wymagań
- współpracę w IW
- testowanie/symulację wymagań.

Narzędzia często wspierają kilka wymienionych powyżej obszarów. Aby zapewnić odpowiednie wsparcie wszystkich zadań IW, można łączyć ze sobą różne narzędzia.

Zdarza się, iż do zarządzania wymaganiami używane są inne narzędzia, takie jak pakiety biurowe lub narzędzia do zgłaszania i śledzenia problemów. Mają one jednak swoje ograniczenia i powinny być używane tylko wtedy, gdy proces IW jest pod kontrolą, a wymagania są uzgodnione i dość stabilne.

7.2 Wprowadzanie narzędzi (L2)

Wybór narzędzia IW nie różni się od wyboru narzędzia jakiegokolwiek innego przeznaczenia. Jeśli chcemy dokonać udanego wyboru, najpierw musimy poprawnie określić cel, kontekst i wymagania [Fugg1993].

Narzędzi można szukać dopiero po określeniu odpowiednich procedur IW i wyborze właściwych technik. Wprowadzanie narzędzi wymaga jasno zdefiniowanych odpowiedzialności IW i procedur. W procesie wprowadzania narzędzia IW istotne są następujące aspekty:

- wszystkie koszty cyklu życia narzędzia, nie tylko koszty licencji
- rozważenie wymaganych zasobów
- wykorzystanie projektów pilotażowych w celu uniknięcia ryzyk
- ocena narzędzia według określonych kryteriów
- odpowiednie przeszkolenie pracowników.

Odniesienia

- [BaCC2015] K. Baxter, C. Courage, K. Caine: Understanding Your Users: A Practical Guide to User Research Methods, 2nd edition. Morgan Kaufmann, Burlington, 2015.
- [BiSp2003] K. Bittner, I. Spence: Use Case Modelling. Pearson Education, Boston, 2003.
- [Bour2009] L. Bourne: Stakeholder Relationship Management: A Maturity Model for Organisational Implementation. Gower Publishing Ltd, Burlington, 2009.
- [CaDJ2014] D. Carrizo, O. Dieste, N. Juristo: Systematizing requirements elicitation technique selection. Information and Software Technology 2014, 56(6): 644–669.
- [Cock2001] A. Cockburn: Writing Effective Use Cases. Addison–Wesley, Boston 2001.
- [Cohn 2004] M. Cohn: User Stories Applied – For Agile Software Development. Addison–Wesley, Boston, 2004.
- [Coop2004] A. Cooper: The Inmates Are Running the Asylum: Why High–Tech Products Drive Us Crazy and How to Restore the Sanity. Que, Indianapolis, 2004.
- [Davi2005] A. M. Davis: Just Enough Requirements Management – Where Software Development Meets Marketing. Dorset House Publishing, Nowy Jork, 2005.
- [Davi1993] A. M. Davis: Software Requirements – Objects, Functions, & States, 2nd edition, Prentice Hall, New Jersey, 1993.
- [DeMa1978] T. DeMarco: Structured Analysis and System Specification. Yourdon Press, Nowy Jork, 1978.
- [Fugg1993] A. Fuggetta: A classification of CASE technology. IEEE Computer 1993, 26 (12): 25–38.
- [GiGr1993] T. Gilb, D. Graham: Software Inspection. Addison Wesley, Boston, 1993.
- [Glin2019] M. Glinz: Requirements Engineering I. Course Notes, Uniwersytet w Zurychu, 2019. <https://www.ifi.uzh.ch/en/rerg/courses/archives/hs19/re-i.html#resources>. Ostatnio odwiedzana: lipiec 2020.
- [Glin2020] M. Glinz: A Glossary of Requirements Engineering Terminology. wersja 2.0. <https://www.ireb.org/en/downloads/#cpre-glossary>. Ostatnio odwiedzana: lipiec 2020.
- [GoFi1994] O. Gotel, A. Finkelstein: An Analysis of the Requirements Traceability Problem. 1st International Conference on Requirements Engineering, Colorado Springs, 1994. 94–101.
- [GoRu2003] R. Goetz, C. Rupp: Psychotherapy for System Requirements. 2nd IEEE International Conference on Cognitive Informatics (ICCI'03), Londyn, 2003. 75–80.
- [GRL2020] Goal oriented Requirement Language. University of Toronto, Canada <https://www.cs.toronto.edu/km/GRL>. Ostatnio odwiedzana: maj 2020..

- [Hare1988] D. Harel: On Visual Formalisms. *Communications of the ACM* 1988, 31 (5): 514–530.
- [HoSch2020] S. Hofer, H. Schwentner: Domain Storytelling — A Collaborative Modeling Method. Available from Leanpub, <http://leanpub.com/domainstorytelling>. Ostatnio odwiedzana: lipiec 2020.
- [HuJD2011] E. Hull, K. Jackson, and J. Dick: *Requirements Engineering*. Springer, 3rd Ed, 2011.
- [ISO29148] ISO/IEC/IEEE 29148: Systems and Software Engineering – Life Cycle Processes – Requirements Engineering, International Organization for Standardization, 2018.
- [ISO19650] ISO 19650: Organization and Digitization of Information about Buildings and Civil Engineering Works, including Building Information Modelling (BIM)– Information Management Using Building Information Modelling – Part 1 and 2, International Organization for Standardization, 2018.
- [ISO25010] ISO/IEC/IEEE25010:2011: Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. International Organization for Standardization, Geneva, 2011.
- [Jack1995] M. A. Jackson: *Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices*. Addison–Wesley, Nowy Jork, 1995.
- [Jack1995b] M. Jackson: The World and the Machine. 17th International Conference on Software Engineering 1995 (ICSE 1995). 287–292.
- [KAA1984] N. Kano et al.: Attractive quality and must–be quality. *Journal of the Japanese Society for Quality Control* 1984, 14(2): 39–48. (in Japanese)
- [KoTh2017] R. Kohavi, S. Thomke: The Surprising Power of Online Experiments – Getting the most out of A/B and other controlled tests. *Harvard Business Review*, Sept–Oct 2017: 74–82.
- [Kuma2013] V. Kumar: *101 Design Methods – A Structured Approach for Driving Innovation in Your Organization*. John Wiley & Sons, Hoboken, 2013.
- [Laue2002] S. Lauesen: *Software Requirements. Styles and Techniques*. Addison–Wesley, Harlow, 2002.
- [Leff2011] D. Leffingwell: *Agile Software Requirements, Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison–Wesley, Boston, 2011.
- [LiOg2011] J. Liedtka, T. Ogilvie: *Designing for Growth: A Design Thinking Tool Kit For Managers*. Columbia University Press, 2011.
- [LiSZ1994] H. Lichter, M. Schneider–Hufschmidt, H. Zullighoven: Prototyping in Industrial Software Projects – Bridging the Gap Between Theory and Practice. *IEEE Transactions on Software Engineering* 1994, 20 (11): 825–832.
- [MFeA2019] D. Méndez Fernández, X. Franch, N. Seyff, M. Felczer, M. Glinz, M. Kalinowski, A. Volgelsang, S. Wagner, S. Bühne, K. Lauenroth: Do We Preach What We

Practice? Investigating the Practical Relevance of Requirements Engineering Syllabi – The IREB Case. *CibSE* 2019: 476–487.

- [Moor2014] C. W. Moore: *The Mediation Process – Practical Strategies for Resolving Conflicts*, 4th edition. John Wiley & Sons, Hoboken, 2014.
- [MWHN2009] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak: *Easy Approach to Requirements Syntax (EARS)*. 17 IEEE International Requirements Engineering Conference (RE'09), Atlanta, Georgia, 2009. 317–322.
- [OleA2018] K. Olsen et al.: *Certified Tester, Foundation Level Syllabus – Version 2018*. International Software Testing Qualifications Board, 2018.
- [OMG2013] Object Management Group: *Business Process Model and Notation (BPMN)*, version 2.0.2. OMG document formal/2013–12–09
<http://www.omg.org/spec/BPMN>. Ostatnio odwiedzana: lipiec 2020.
- [OMG2017] Object Management Group: *OMG Unified Modeling Language (OMG UML)*, version 2.5.1. OMG document formal/2017–12–05.
<https://www.omg.org/spec/UML/About-UML/>. Ostatnio odwiedzana: lipiec 2020.
- [OMG2019] Object Management Group: *OMG Systems Modeling Language (OMG SysML™)*, Version 1.6. OMG Document formal/19–11–01.
<https://www.omg.org/spec/SysML/>. Ostatnio odwiedzana: styczeń 2022.
- [Osbo1979] A. F. Osborn: *Applied Imagination*, 3rd revised edition. Charles Scribner's Sons, New York, 1979.
- [RoRo2012] S. Robertson and J. Robertson: *Mastering the Requirements Process*, 3rd edition. Addison–Wesley, Boston, 2012.
- [Rupp2014] C. Rupp: *Requirements–Engineering und Management*, 6. Auflage. Hanser, München, 2014. (in German).
- [Sdsc2012] Stanford d.school: *An Introduction to Design Thinking*. Hasso Plattner Institute of Design, Stanford, 2012. <https://dschool-old.stanford.edu/groups/designresources/wiki/36873>. Ostatnio odwiedzana: lipiec 2020.
- [vLam2009] Axel van Lamsweerde: *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Chichester: John Wiley & Sons, 2009.
- [Vole2020] Volere: *Requirements Resources*. <https://www.volere.org>. Ostatnio odwiedzana: lipiec 2020.
- [WiBe2013] K. Wiegers and J. Beatty: *Software Requirements*, 3rd edition. Microsoft Press, Redmond, 2013.