

Celem jest określenie UCP dla systemu wskazanego przez prowadzącego. Najpierw cała grupa uzgadnia wartości czynników środowiskowych i technicznych (nie mamy rzeczywistych danych, więc większość należy „rozsądnie zgadnąć”). Dalsze prace należy wykonać w zespołach dwuosobowych. Zespoły nie powinny się ze sobą komunikować, aby na koniec zajęć móc porównać wyniki swoich szacunków.

Model użyty do estymacji zasobów potrzebnych do wykonania systemu jest także podstawą zaliczenia laboratorium, więc jego dobra znajomość jest jak najbardziej pożyteczna.

## Metoda Use Case Points<sup>1</sup>

Metoda UCP (punktów przypadków użycia) została opracowana w połowie lat dziewięćdziesiątych ubiegłego stulecia w Rational Software równolegle z pierwszą wersją UML. Ma na celu określenie złożoności (wielkości) projektu, a w efekcie jego pracochłonności, na podstawie modelu przypadków użycia. Nasuwa się pewne podobieństwo do metody punktów funkcyjnych, jednak UCP jest znacznie bardziej wysokopoziomowa i pozwala na wykonanie szacunku we wczesnych fazach opracowania projektu (model UC jest podstawą prac nad projektem, a punkty funkcyjne wymagają informacji o modułach projektu, wejściach, wyjściach, operacjach IO, zapytaniach, plikach wewnętrznych i zewnętrznych).

Ważnym założeniem metody UCP jest opracowanie wszystkich przypadków użycia na dokładnie tym samym poziomie szczegółowości. Poziom ten nie jest dowolny – zgodnie z klasyfikacją Alistaira Cockburna powinien to być poziom średni, tj. *user goal* – zorientowany na cele użytkownika, czyli właśnie taki, z jakim mieliśmy do czynienia podczas laboratoriów (wszystkie pięć poziomów to: *very high summary*, *summary*, *user goal*, *subfunction*, *tool low*).

Czynniki uwzględniane przez metodę UCP to:

- aktorzy,
- przypadki użycia
- czynniki złożoności środowiska
- czynniki złożoności technicznej.

## Aktorzy

Wyróżniamy trzy typy aktorów, uwzględniając ich złożoność:

<i>Klasa</i>	<i>Opis</i>	<i>Waga</i>
prosty	System komunikujący się przez zdefiniowane API	1
średni	System komunikujący protokół sieciowy lub inny interakcyjny, np. terminal znakowy	2
złożony	Użytkownik wymagający GUI	3

Każdemu typowi przypisana jest waga. Zliczamy aktorów każdego typu i mnożymy przez wagę. Uzyskujemy w ten sposób wartość *unadjusted actors weights* (UAW) – sumę iloczynów liczb aktorów danego typu i wagi typu:

---

<sup>1</sup> Opracowane w oparciu o *Applied Use Cases* (G. Schneider, J.P. Winters)

$$UAW = n_{\text{prosty}} w_{\text{prosty}} + n_{\text{średni}} w_{\text{średni}} + n_{\text{złożony}} w_{\text{złożony}}$$

## Przypadki użycia

Podobnie jak w przypadku aktorów, przypadki użycia klasyfikujemy względem klas złożoności:

Klasa	Opis	Waga
prosty	Do 3 transakcji	5
średni	4 do 7 transakcji	10
złożony	Więcej niż 7 transakcji	15

Transakcja jest rozumiana jako krok przypadku użycia (jest to pewne uproszczenie, ale dobrze się sprawdza), jest więc jednostkową komunikacją aktora z systemem (parą „pytanie-odpowiedź”, czasami w jednej transakcji możemy uwzględnić kilka takich par). Widać stąd, dlaczego musieliśmy wprowadzić ograniczenie i ujednolicenie poziomu szczegółowości przypadków użycia.

Alternatywą dla powyższej oceny złożoności przypadków użycia jest oparcie się o liczbę klas w modelu analizy:

Klasa	Opis	Waga
prosty	Do 4 klas	5
średni	5 do 10 klas	10
złożony	Więcej niż 10 klas	15

Należy wybrać jedną z powyższych możliwości i stosować ją konsekwentnie przez cały proces szacowania (a najlepiej we wszystkich szacowaniach różnych projektów, dla uniknięcia nieporozumień).

Mnożąc sumując iloczyny liczby przypadków danej klasy przez wagę klasy uzyskujemy *unadjusted use case weight* (UUCW):

$$UUCW = n_{\text{prosty}} w_{\text{prosty}} + n_{\text{średni}} w_{\text{średni}} + n_{\text{złożony}} w_{\text{złożony}}$$

## Czynniki złożoności środowiskowej

Czynniki złożoności środowiskowej (*environmental complexity factors*, ECF) oceniamy w przedziale [0; 5] i mnożymy przez ich wagę. Uwzględniane ECF to:

Czynnik	Opis	Waga
E1	Zaznajomienie członków zespołu z projektem, metodyką, procesem	1.5
E2	Doświadczenie zespołu w tworzeniu systemów danego typu i w dziedzinie problemu	0.5
E3	Doświadczenie zespołu w tworzeniu aplikacji obiektowych	1.0
E4	Kwalifikacje/doświadczenie głównego analityka	0.5

E5	Motywacja członków zespołu	1.0
E6	Stabilność wymagań	2.0
E7	Udział pracowników zewnętrznych/w niepełnym wymiarze zatrudnienia	-1.0
E8	Trudność języka programowania	-1.0

Obliczamy czynnik środowiskowy **EF** (*environmental factor*) zgodnie z następującym wzorem:

$$EF = 1.4 - 0.03(E_1 w_1 + \dots + E_8 w_8)$$

### Czynniki złożoności technicznej

Stosujemy metodę analogiczną do czynników środowiskowych i oceniamy poszczególne czynniki techniczne (*technical complexity factors*, TCF) w przedziale [0; 5] mnożąc przez wagi. Proszę zwrócić uwagę do podobieństwa do wymagań niefunkcjonalnych:

Czynnik	Opis	Waga
T1	System rozproszony	2.0
T2	Wydajność (czas reakcji, przepustowość)	1.0
T3	Sprawność działania dla użytkownika końcowego	1.0
T4	Złożone przetwarzanie wewnętrzne	1.0
T5	Wymaganie ponownej używalności kodu	1.0
T6	Łatwość instalacji	0.5
T7	Łatwość użytkowania	0.5
T8	Przenośność	2.0
T9	Łatwość wprowadzania zmian	1.0
T10	Współbieżność	1.0
T11	Specjalne zabezpieczenia i mechanizmy ochrony dostępu	1.0
T12	Zależność od systemów zewnętrznych (np. udostępnianie im danych)	1.0
T13	Konieczność specjalnego przeszkolenia personelu	1.0

Obliczamy czynnik techniczny **TF** (*technical factor*) ze wzoru:

$$TF = 0.6 + 0.01(T_1 w_1 + \dots + T_{13} w_{13})$$

### Złożoność projektu

Mając zebrane wszystkie wartości, możemy obliczyć **UUCP** (*unadjusted use case points*):

$$UUCP = UAW + UUCW$$

A następnie zastosować czynniki środowiskowe i techniczne, dla obliczenia punktów przypadków użycia, czyli **UCP** (*use case points*):

$$UCP = UUCP * EF * TF$$

Dalej, możemy określić pracochłonność projektu przyjmując odpowiedni czynnik produktywności (*productivity factor*, PF), według Gustava Karnera<sup>2</sup> określony jako 20 roboczogodzin na 1 UCP:

$$\text{pracochłonność} = UCP * PF$$

---

2 *Use Case Points – Resource Estimation for Objectory Projects* (G. Karner)