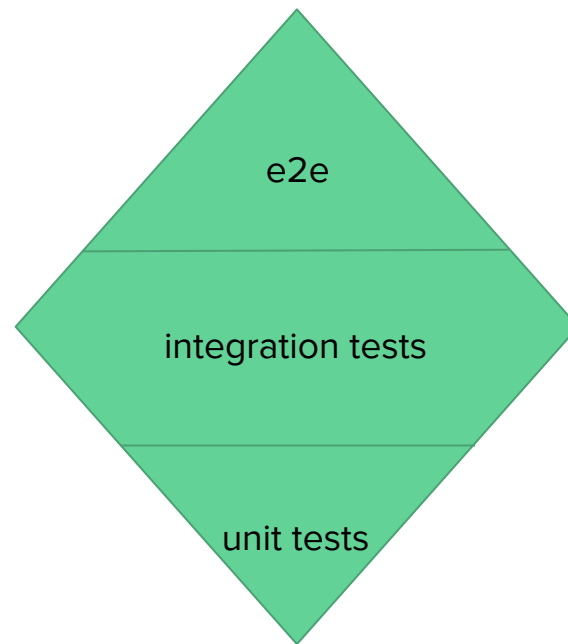
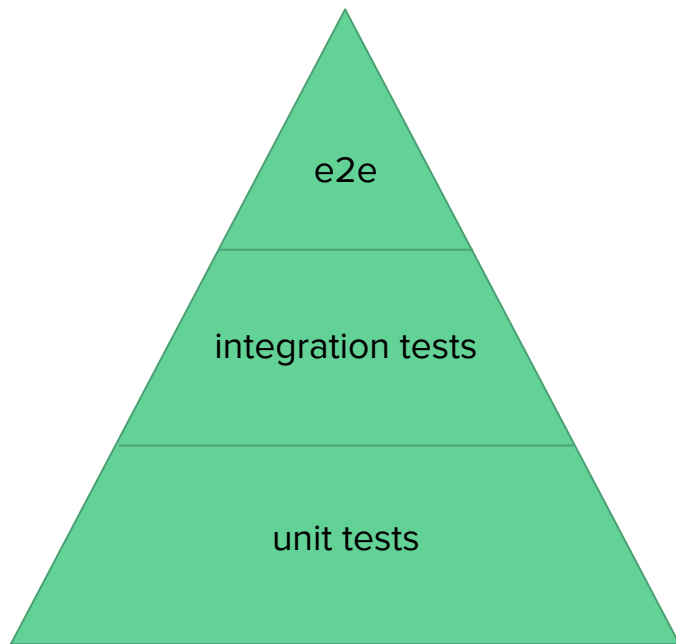


**e2e**

# Agenda

1. Poziomy testów
2. Jak testować manualnie?
3. Automatyzacja testów e2e: wzorce i zasady
4. Zadanie domowe

# 1. Poziomy testów



## 2. Jak testować manualnie

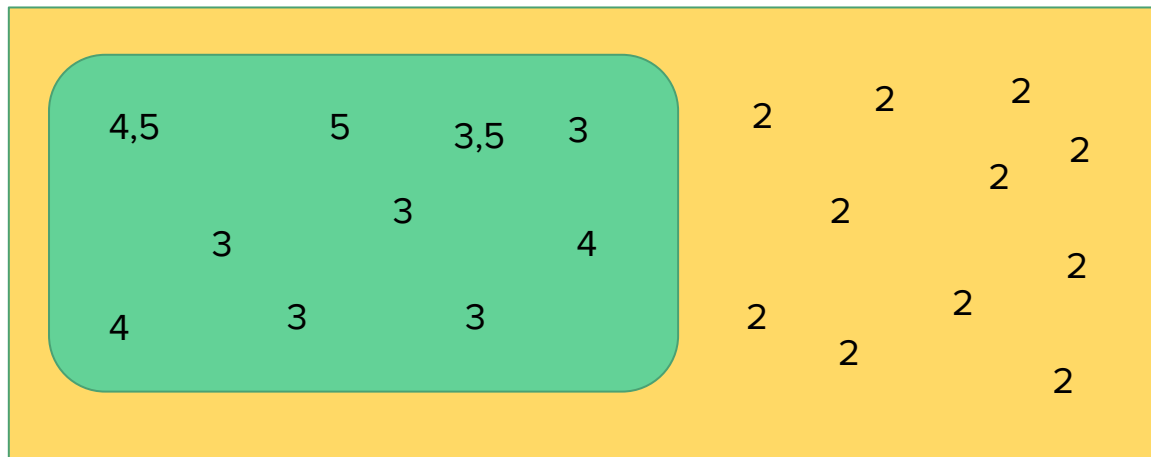
happy path

co może pójść nie tak?



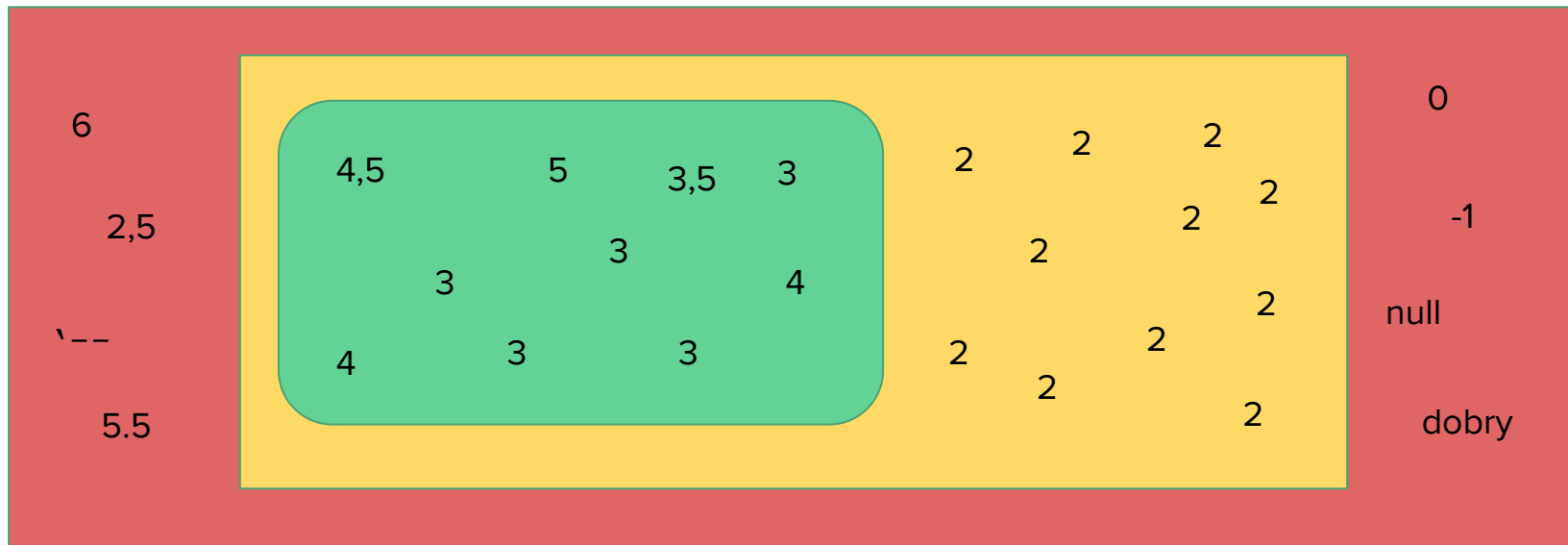
## 2. Jak testować manualnie

analiza wartości brzegowych  
klasy równoważności



## 2. Jak testować manualnie

analiza wartości brzegowych  
klasy równoważności



## 2. Jak testować manualnie

ćwiczenie\*

## 2. Jak testować manualnie

ścieżki użytkownika i wymagania biznesowe

- *jako X chcę zrobić Y, aby Z*: użytkownik, cel, kontekst
- np. jako użytkownik posiadający konto w serwisie Trello chcę się do niego zalogować, aby zobaczyć swój panel użytkownika



## 2. Jak testować manualnie

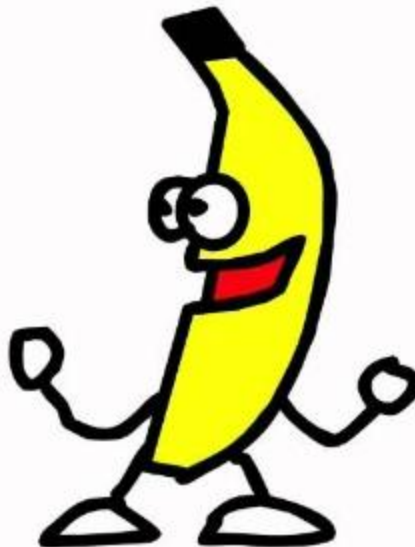
- wymagania wstępne: użytkownik nie jest zalogowany, użytkownik posiada login w serwisie (login: trellouser, hasło: supersecret)
- przebieg testu:
  - a. wejdź na stronę <https://trello.com/login>
  - b. w polu identyfikator wpisz: trellouser
  - c. w polu hasło wpisz: supersecret
  - d. kliknij Log in

## 2. Jak testować manualnie

- oczekiwany wynik: użytkownik zostaje zalogowany i przekierowany na swój panel użytkownika o adresie <https://trello.com/trellouser/boards>

## 2. Jak testować manualnie

ćwiczenie



**IT'S PEANUT-BUTTER-JELLY TIME**

### 3. Automatyzacja testów e2e: wzorce i zasady

wzorzec: “brak wzorca”

```
cy.visit(`https://trello.com/login`);  
  
cy.get(`#user`).type(`trellouser`);  
cy.get(`#password`).type(`supersecret`);  
cy.get(`#login`).click();  
  
cy.url().should('eq', 'https://trello.com/trellouser/boards');
```

### 3. Automatyzacja testów e2e: wzorce i zasady

wzorzec: “brak wzorca”

problemy:

- co jeśli zmieni się url do loginu?
- co jeśli będziemy mieć 10 testów sprawdzających logowanie w różnych wariantach?
- co jeśli zmienią się selektory?
- co jeśli będziemy testować na różnych środowiskach? (prod, integracja, stage)

### 3. Automatyzacja testów e2e: wzorce i zasady

wzorzec: “brak wzorca”

plusy:

- zastosowanie zasady setup - execution - assertion
- szybko się pisze (na początku...)

kiedy?

- bardzo małe i proste aplikacje / widoki

### 3. Automatyzacja testów e2e: wzorce i zasady

wzorzec Page Object Pattern: selektory i wzorce stron

```
../page_objects/login/login.po.js
```

```
export default {  
  selectors: {  
    inputs: {  
      username: '#user',  
      password: '#password'  
    },  
    buttons: {  
      login: '#login'  
    }  
  }  
}
```

```
../page_objects/base/abstract-page.js
```

```
import envData from '../data/env.json';  
  
export default class AbstractPage {  
  constructor() {  
    this.domain =  
envData.env[global.envTest].URL;  
    this.pagePath = '';  
  }  
  /* place for shared page methods */  
}
```

### 3. Automatyzacja testów e2e: wzorce i zasady

wzorzec Page Object Pattern: klasy stron

```
../page_objects/login/login.js

import PageObject from './login.po';

export default class Login extends AbstractPage {
  constructor() {
    super();
    this.pagePath = '/login';

    this.selectors = { ...PageObject.selectors };
  }
  /* ... */
}
```



### 3. Automatyzacja testów e2e: wzorce i zasady

wzorzec Page Object Pattern: klasy stron

```
../page_objects/login/login.js

export default class Login extends AbstractPage {
  /* ... */
  login = (user, password) => {
    cy.get(this.selectors.inputs.username).type(user);
    cy.get(this.selectors.inputs.password).type(password);
  }
}
```

### 3. Automatyzacja testów e2e: wzorce i zasady

wzorzec Page Object Pattern: specyfikacje testów

```
../specs/login.js

import Login from '../page_objects/login/login'
import loginData from '../data/loginData.json'

describe('Log-in form tests -- successful', function() {
  const Login = new Login();

  beforeEach(() => {
    Login.open();
  })
  /* ... */
})
```

### 3. Automatyzacja testów e2e: wzorce i zasady

wzorzec Page Object Pattern: specyfikacje testów

```
../specs/login.js

/* ... */

it('Redirects you to the user dashboard if you provide the correct
credentials', function() {
  Login.login(loginData.userCorrect, loginData.userIncorrect);
  cy.url().should('eq', envData.env[global.envTest].URL +
loginData.userCorrect + '/boards');
})
}
```

### 3. Automatyzacja testów e2e: wzorce i zasady

wzorzec: Page Object Pattern

plusy:

- zastosowanie zasady setup - execution - assertion
- korzystamy z walorów pisania obiektowego

minusy:

- bardzo. dużo. pisania. obiektów.

kiedy?

- złożone aplikacje z różnorodnymi komponentami i akcjami użytkownika

### 3. Automatyzacja testów e2e: wzorce i zasady

wzorzec: coś pomiędzy?

- selektory i dane testowe wyabstrahowane do osobnych plików
- zamiast tworzyć obiekty stron skupiamy się wspólnych akcjach umieszczanymi w utilsach (ew. kolejnych plikach pomocniczych)

plusy:

- ciągle jest DRY
- nie klepiemy tylu obiektów

minusy:

- nie każdy się odnajdzie w życiu poza obiektami

### 3. Automatyzacja testów e2e: wzorce i zasady

zasady:

- automatyzuj znając kroki i rezultat końcowy
- preferuj rzadko zmieniające się selektory:
  - id > name > class > xpath
- powtarzalne metody/akcje definiuj w utilsach/commandsach/itp
- staraj się emulować zachowania użytkownika
  - klikanie checkboxów zamiast zmiany stanu
  - wypełnianie inputów zamiast zmiany wartości

## 4. Zadanie domowe

Zautomatyzuj testy strony przygotowanej przez siebie w ramach warsztatów z a11y. :)

Alternatywnie:

Zapoznaj się z tą stroną<sup>\*</sup><sub>—</sub> i napisz testy e2e dla dowolnie wybranej funkcjonalności, uwzględniając *happy path* i przypadki negatywne.