
LABORATORIO APPLICAZIONE SW E SICUREZZA INFORMATICA

ROBERTO BERALDI



PERCHE' PROGETTI SOFTWARE FALLISCONO?

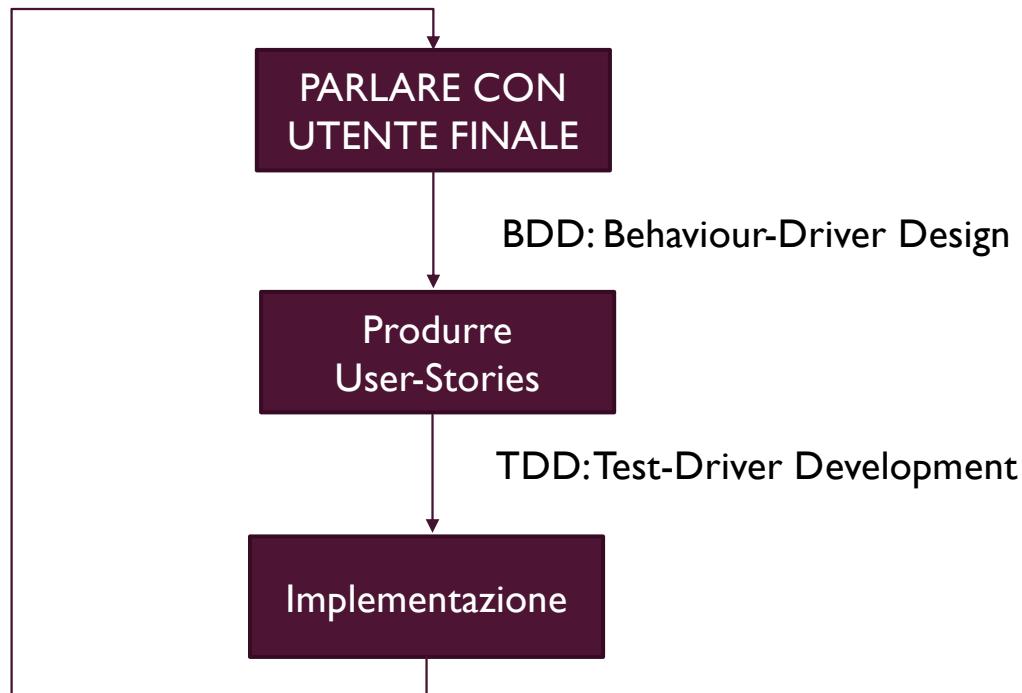
- Non fare quello che vogliono i clienti
- Oppure i progetti sono in ritardo
- O oltre il budget
- O difficile da mantenere ed evolvere
- O tutto quanto sopra
- **Agile** è una metodologia che cerca di aiutare ad evitare questi problemi

AGILE

- Lavorare a stretto contatto, continuamente con le parti interessate per sviluppare requisiti e test
 - Utenti, clienti, sviluppatori, programmatore di manutenzione, operatori, project manager,
- Gestisci prototipo durante la distribuzione di nuove funzionalità ogni iterazione
 - In genere ogni 1 o 2 settimane
- Verifica con parti interessate come procedure, per **convalidare** che si stia costruendo la cosa giusta

ITERAZIONE AGILE

< 2 settimane...

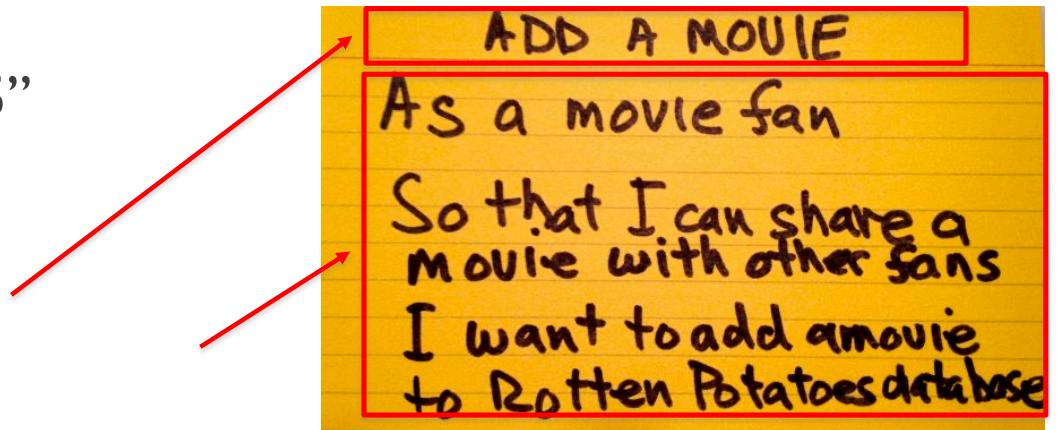


BDD

- BDD vuol dire porsi domande sul **comportamento** atteso dell'applicazione da parte dell'utente, prima e durante lo sviluppo, e ciò al fine di ridurre le incomprensioni
- Lo scopo di questo approccio è avere una **convalida** della correttezza dell'app rispetto alle attese (**acceptance test**)
- La **verifica** è piuttosto un aspetto tecnico, nel senso che la verifica può essere positive anche per applicazioni non validate (ossia che nn fanno ciò che l'utente si aspetta)
- I Requisiti sono scritti in termini di **USER STORIES**
 - Descrizioni dell'uso dell'app
- BDD si concentra sul comportamento dell'app rispetto all'implementazione dell'app

USER-STORIES

- 1-3 frasi nel linguaggio quotidiano
- Si adatta alla scheda indice da 3" x 5"
- Scritto da/con il cliente
- Nome della **feature**
- **AS** [tipo di stakeholder],
- **SO THAT** [posso raggiungere qualche obiettivo],
- **I WANT TO** [fare qualche compito]



PUNTI DIVISTA DIFFERENTI

- *See which of my friends are going to a show*
 - As a theatergoer
 - So that I can enjoy the show with my friends
 - I want to see which of my Facebook friends are attending a given show
- *Show patron's Facebook friends*
 - As a box office manager
 - So that I can induce a patron to buy a ticket
 - I want to show her which of her Facebook friends are going to a given show

BACKLOG

- I sistemi reali hanno dell'ordine delle centinaia di user-story
-
- **Backlog:** User-stories non ancora completate
- Si da priorità storie più important mettendole in cima alla lista
-
- Raggrupparle in modo che esse corrispondano alle versioni SW nel tempo

SPIKE

- Breve indagine sulla tecnica o sul problema
- Ad esempio, spike sugli algoritmi di raccomandazione
- Sperimentare degli algoritmi
- Limitato nel tempo
- Al termine, buttare via il codice (!)
- E riscriverlo ‘bene’ da capo ..

COME CREARE LE USER STORIES?

- Right size?
- Not too hard?
- Is worthwhile?

SMART USER STORIES

- **S**pecific
- **M**easurable
- **A**chievable (ideally, implement in 1 iteration)
- **R**elevant
- **T**imeboxed (know when to give up)

USER-STORY: SPECIFICA E MISURABILE

- US specifica ossia riferita ad un particolare scenario
- Testabile implica individuare uno o più inputs e relativi risultati attesi
-
- Contro-esempio: " UI should be user-friendly "
- Esempio: **Given/When/Then.**
- **Given** some specific starting condition(s),
- **When** I take specific action X,
- **Then** one or more specific thing(s) should happen

USER-STORY: ACHIEVABLE

- Complete in una iterazione agile
 - Se non è possibile in 1 iterazione, dividerle in un sottoinsieme di storie
 - Puntare sempre al codice alla fine dell'iterazione
-
- Se < 1 storia per iterazione, è necessario migliorare la stima dei punti per storia

COME SAPERE QUANTO TEMPO CI VUOLE?

- Ogni user-story viene valutata dal team di sviluppo in anticipo usando una semplice scala intera e dando loro un peso, ad esempio
 - 1 per storie semplici,
 - 2 per storie medie,
 - 3 per storie molto complesse
 - ..O anche serie di Fibonacci
- Ogni US riceve così un numero di punti
- Velocità del team: numero medio di punti / settimana



USER STORY: BUSINESS VALUE

- Analizzare il perche' si vuole una particolare user story e assegnarle un valore

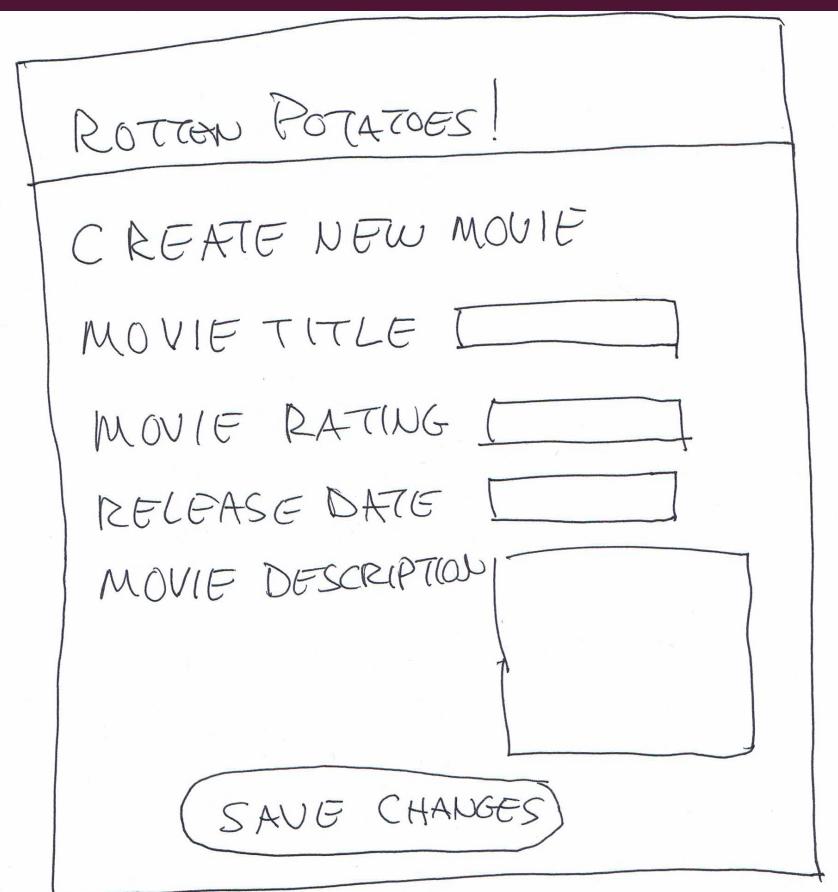
USER STORY:TIME-BOXED

- Interrompere l'implementazione delle UA quando si supera il budget di tempo
- Rinunciare o dividerla in storie più piccole o riprogrammare ciò che rimane annullato

USER STORY E USER INTERFACE

- User stories quasi sempre necessitano dell'interfaccia utente(UI)
- In questi casi conviene presentare la versione UI ai clienti, prima che l'app sia realizzata
- La UI è mostrata in modo stilizzato come **mockup**, approssimazione della UI finale
- Mockup possono essere a ‘bassa fedeltà’ o alta fedeltà a seconda se rispecchiano a meno l’interfaccia finale
- Lo scopo principale è mostrare l’intenzione della UI, ossia l’idea

LO-FI UI (LOW FIDELITY)



(From *Engineering Software as a Service* © by
Armando Fox and David Patterson, used
with permission)

→ C ⌘ wireframepro.mockflow.com/editor.jsp?editor=off&publicid=Mfe3d2fc6dfe81b057d721916916eb3621615982833717&projectid=D7dfc537663... ☆ G ↗ 🔍 ⏪ Paused ⋮

Project Edit Review >

Web

Cmd+K to Search

Browser frame Text

Rectangle Circle

Horizontal line Image

Icon library Button

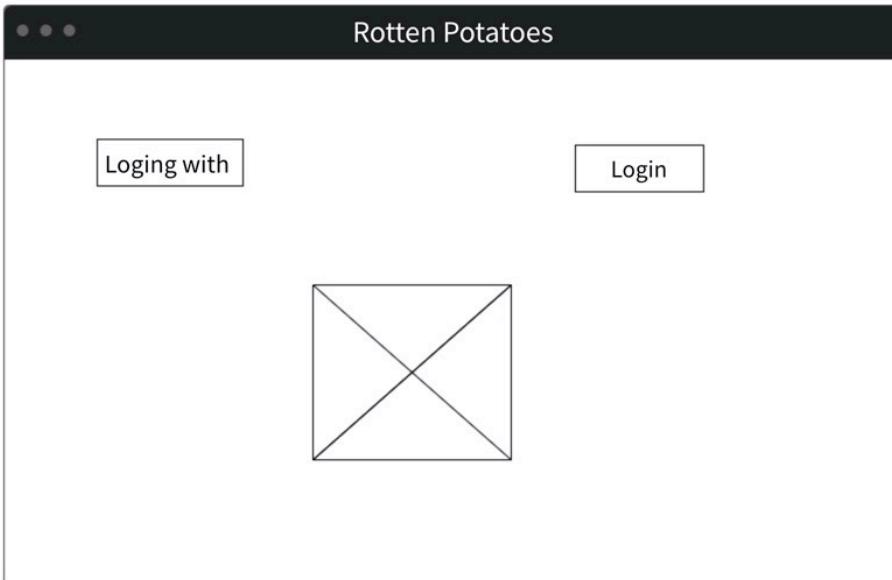
Label... Ad

Input

Project ES RE Z-UP

Rotten Potatoes

Logging with Login



The screenshot shows a wireframe interface for a login screen. The title bar says "Rotten Potatoes". The main content area contains a large square with a diagonal cross. Above the square are two boxes: "Logging with" on the left and "Login" on the right. The interface includes a sidebar with various UI element icons like browser frames, text, rectangles, circles, lines, and buttons.

The screenshot shows the NinjaMock web-based mockup tool interface. At the top, there's a dark header bar with a progress bar and a "Paused" button. Below it is a red banner with the "NinjaMock" logo. The main workspace is titled "Project 'Untitled project'" and contains a white canvas with a "Heading 1" element. On the left, the "Elements" panel lists basic shapes like "Label" and "Link". The "Outliner" panel shows a hierarchical structure with sections for "Logo", "1. Item 1", "2. Item 2", and "1.1 Sub-item 1". The "Pages" panel shows a single page named "Home". The "Properties" panel on the right is open, showing settings for the selected "Label" element, including text ("Label"), font ("TomsHandw..."), size ("22"), color ("#000000"), and layout options like width ("34") and height ("19"). The bottom navigation bar includes links for "upgrade", "feedback", "video", and "shortcuts".

ninjamock.com/Designer/Workplace/145362295/Page1

NinjaMock Project "Untitled project"

Paused

export undo redo comments snap link preview my projects log out

Search element (F3)...

Basic

Label Link

Heading 1

Label

Heading 1

Heading 2

Heading 3

Logo

1. Item 1

1.1 Sub-item 1

2. Item 2

1.1 Sub-item 1

2. Item 2

Pages

Default group

upgrade feedback video shortcuts

Search property (Shift+F3)...

Label

Text: Label

Scale with text

Multiline

Font

Font family: TomsHandw...

Font size: 22

Font color: #000000

Line height: 1.1

Page linking

Page link: None

Appearance

Layout

Width: 34

Height: 19

Left: 439

Top: 321

Anchor: Left Right Top Bottom

ESEMPIO DI MOCKUP DI PROGETTI PASSATI

The mockup displays a web browser window with the following content:

League details

Dettagli

Montepremi

Primo	400
Secondo	200
Terzo	100

Status: Aperta

Partecipanti

Gioc1
Gioc2
Gioc3

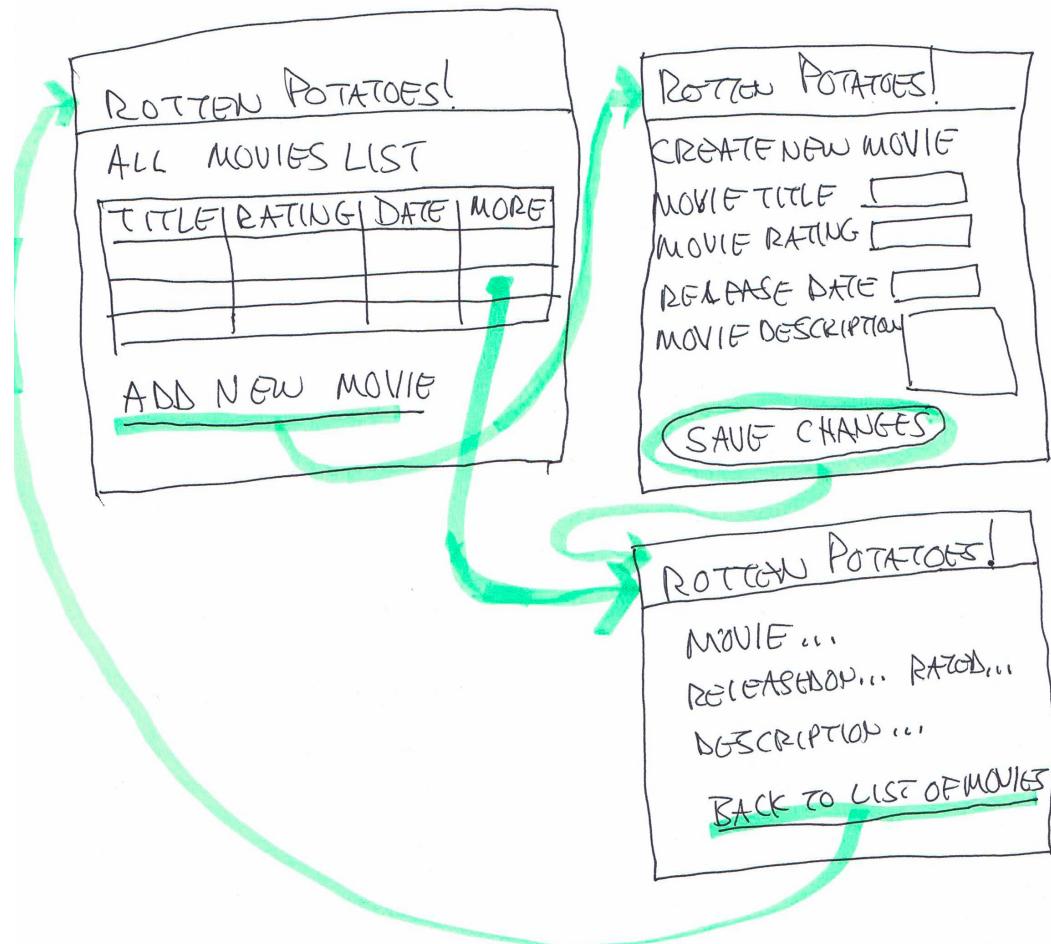
Join

[Back Home](#)

The browser's address bar shows the URL <http://localhost:3000/homes/1>.

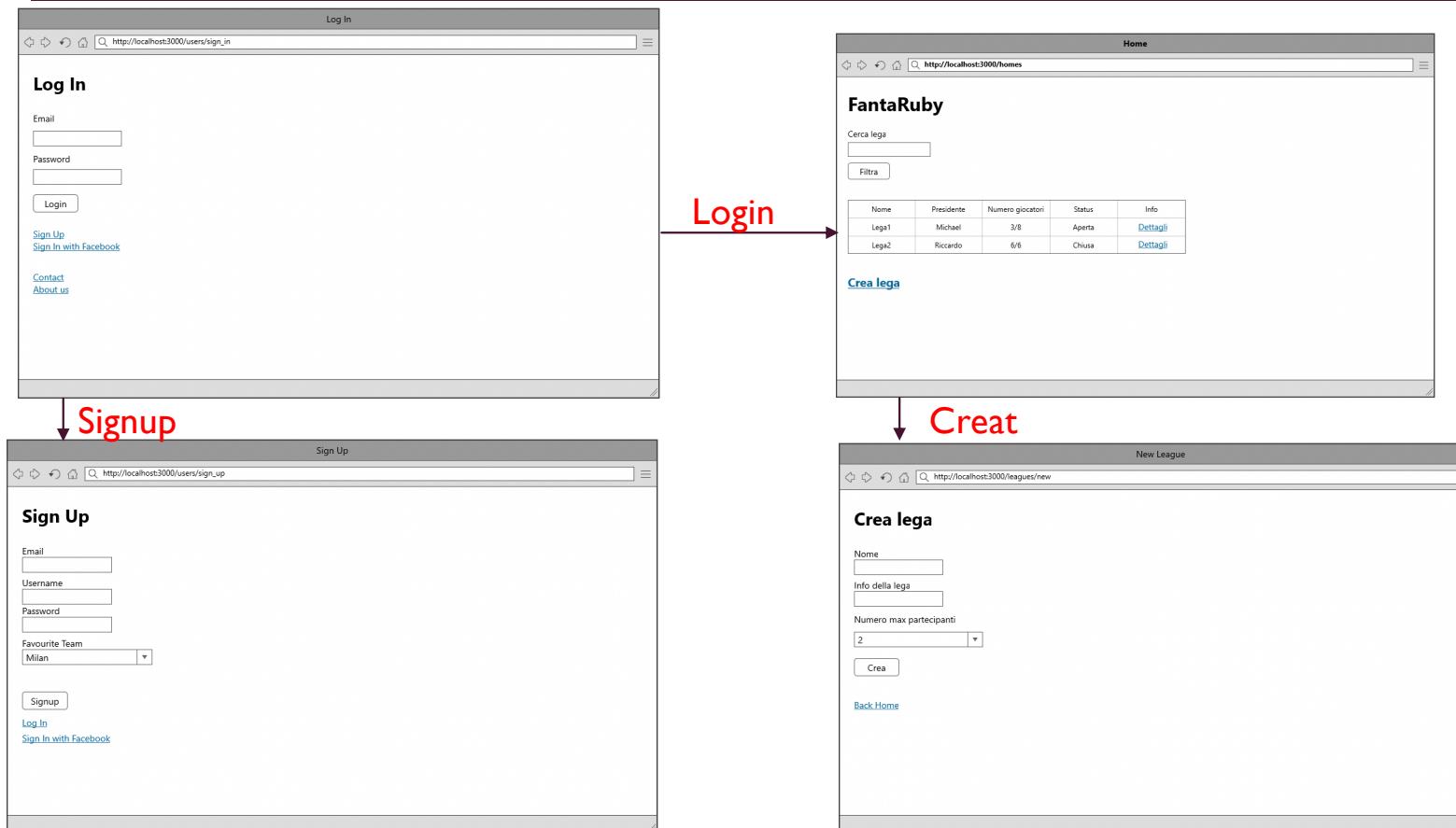
STORYBOARD

- **Storyboard:** Relazione fra UI indotta l'utilizzo di un utente
- Come le scene di un film, ma non lineare perchè dipende da come l'utente interagisce con la app
-



(Figure 4.4, *Engineering Long Lasting Software*
by Armando Fox and David Patterson,
Alpha edition, 2012.)

ESEMPIO DI STORYBOARD



ESEMPIO DI LISTA DI USER STORIES DI UN PROGETTO

User Stories

Unregistered User

1. As an Unregistered User I want to sign up with my e-mail so that I can become a User.
2. As an Unregistered User I want to login with Facebook so that I can become a User.
3. As an Unregistered User I want to have a page so that I can get info about the application.
4. As an Unregistered User I want to have a page so that I can get info about contacts.
5. As an Unregistered User I want to have a page so that I can get soccer's teams info.

User

6. As a User I want to have settings so that I can login with two factors.

USER STORIES E BEHAVIOUR DRIVEN DESIGN (BDD)

- Una **user story** è una narrazione astratta di come l'utente userà l'applicazione,
- Una user story utilizza una o più **features** dell'applicazione
- Per ‘utilizzare’ una user story si definisce uno **scenario** concreto nel quale si userà la feature
- Questo utilizzo della feature supporta la filosofia di design detta **BDD**

DALLE USER STORY ALLE FEATURES

- **AS A MOVIEGOER**
- **I WANT TO ADD A MOVIE**
- **SO THAT ...**

Feature:ADD A MOVIE

CUCUMBER SPECIFICATION

Feature: User can manually add movie

1 Feature

Scenario: Add a movie

≥1 Scenarios / Feature

```
Given I am on the RottenPotatoes home page
When I follow "Add new movie"
Then I should be on the Create New Movie page
When I fill in "Title" with "Men In Black"
And I select "PG-13" from "Rating"
And I press "Save Changes"
Then I should be on the RottenPotatoes home page
And I should see "Men In Black"
```

3 to 8 Steps / Scenario

AUTOMAZIONE DEI TEST DI INTEGRAZIONE

1. **Given** steps represent state of world before event: preconditions
2. **When** steps represent event
 - e.g., simulate user pushing a button
3. **Then** steps represent expected postconditions; check if true
4. / 5. **And** & **But** extend previous step

These are all aliases for same method

CUCUMBER

- User story → feature
- Feature: (almeno) uno scenario che mostra come la feature è usata
 - *happy path & sad path* scenarios
 - features/*.feature
- Scenario: 3 - 8 steps
- Step definitions:
 - Ruby code to test steps
 - `features/step_definitions/*_steps.rb`