factor10

DDD ♥ e2e UI test

Andreas Cederström
andreas.cederstrom@factor10.com

Jimmy Nilsson
jimmy.nilsson@factor10.com

Who are we?

Andreas Cederström

- → Software consultant at factor10
- → Passionate about automation, testing pair/mob programming

Jimmy Nilsson

- → Coding architect and CEO of factor10
- → Author of the second DDD-book
- → Passionate about DDD, XP and the business of our clients

Agenda

factor10

- → Putting e2e UI tests into a DDD context
- → Introduction to UI test and Cypress
- → Lab 1: *up and running*

- → Intro to the domain
- → Syntax and tooling in Cypress (+ live kod av ett testfall)
- → Lab 2: *test after*

- → Ubiquitous Language (Domain expert + BDD -> UI test first)
- → e2e UI test first
- → Lab 3: *test first*

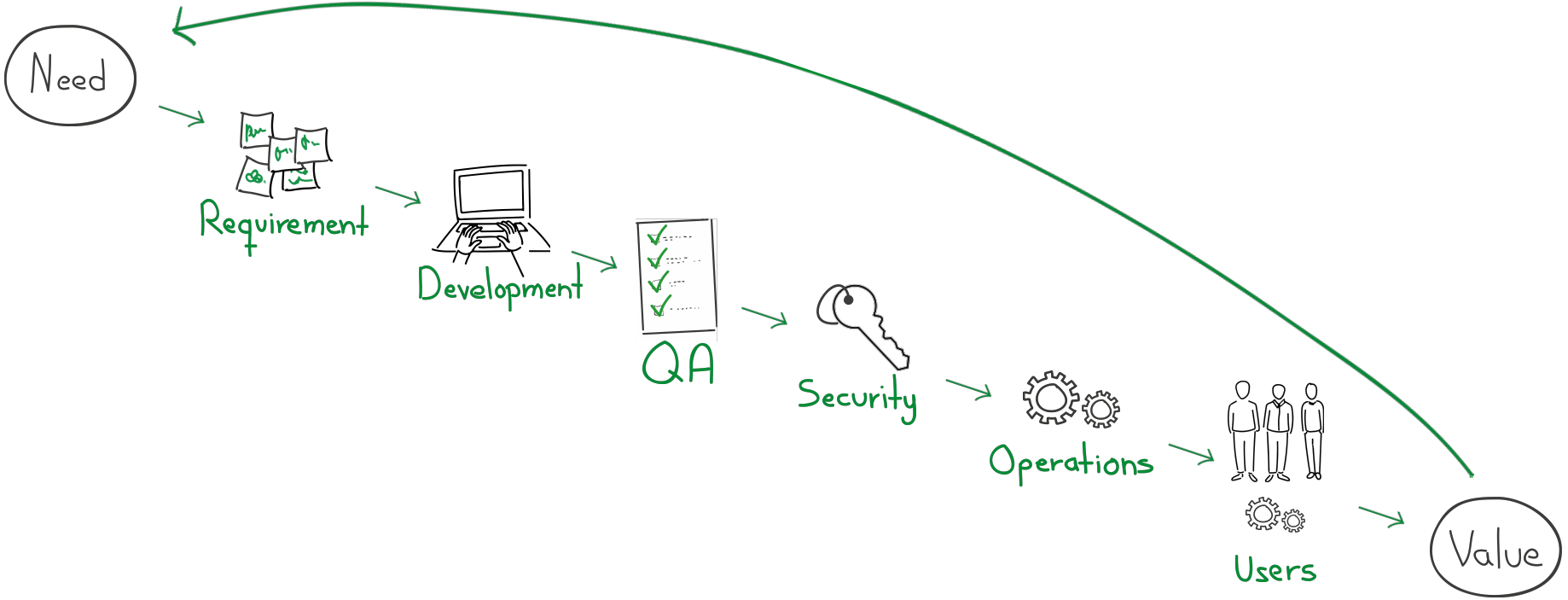
- → Reflections, lessons learned
- → Next steps

factor10

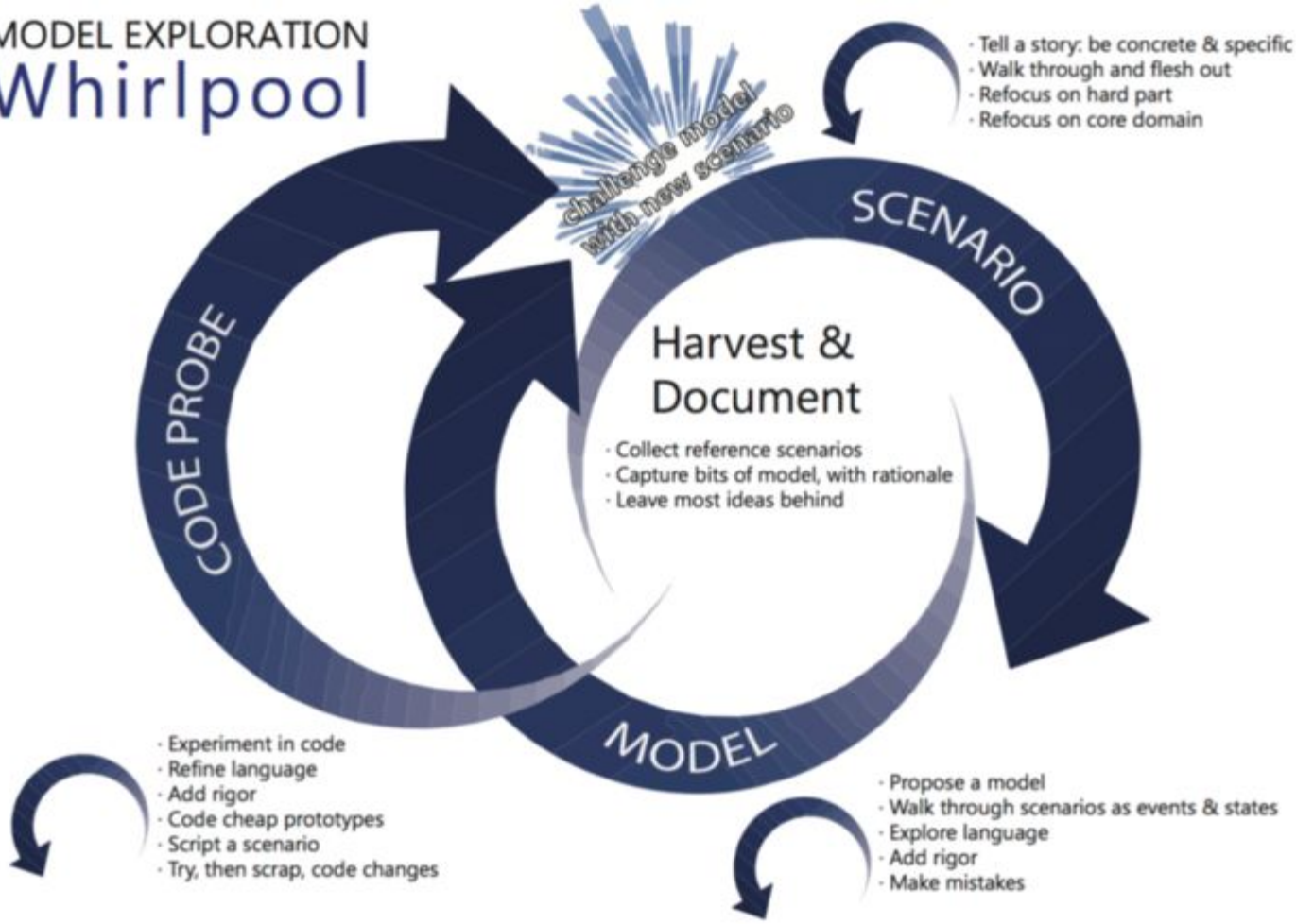
Putting e2e UI tests into a DDD context

Traditional process

factor10



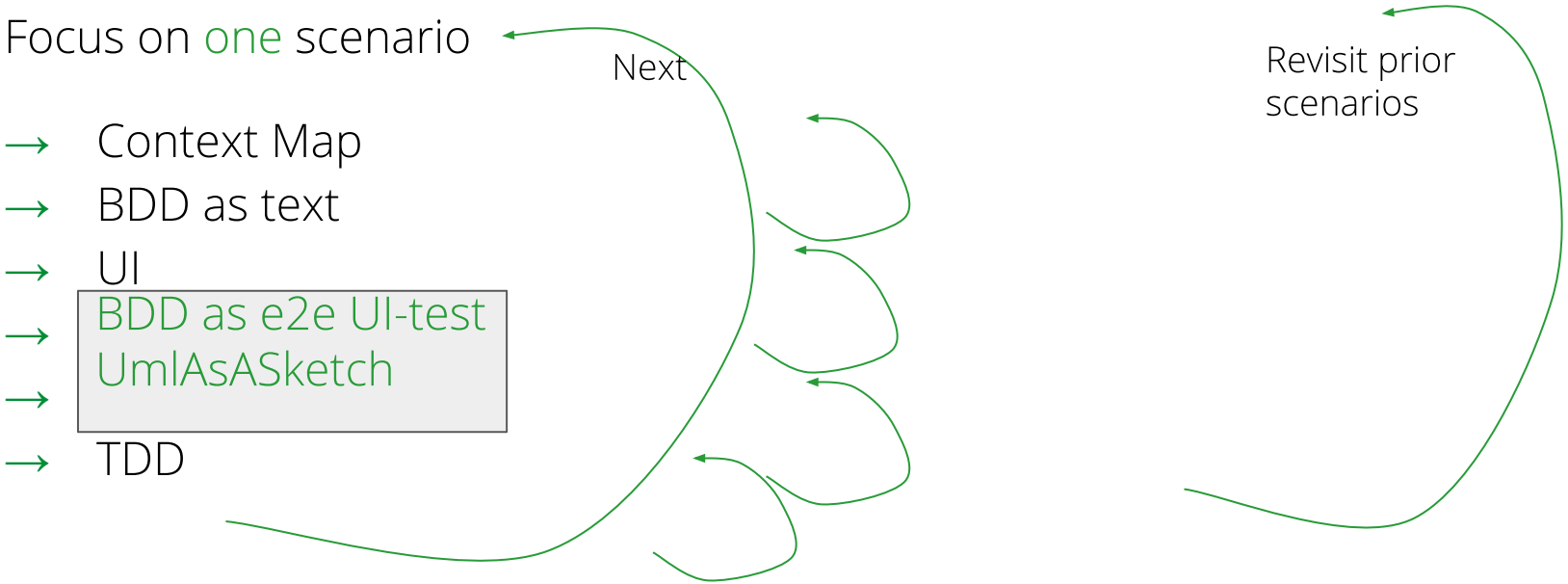
MODEL EXPLORATION
Whirlpool



domainlanguage.com/ddd/whirpool

factor10

Something like this, in collaboration?



Make decisions for now and move on.
Expect to be wrong and change when you understand better.

factor10

Intro to UI test and Cypress

factor10

Definition

What do we mean when we say *end-to-end UI test*?

→ It is an automated test which interacts with a system via the systems user interface.

→ System under test is as close to a production environment as possible including storage and similar dependencies.

factor10

What is Cypress

- → Test automation at GUI level
- → A test runner
- → Web based
- → Javascript
- → Familiar if you are used to Jasmine, Jest or Mocha
- → Fully integrated with browser

factor10

Lab 1
up and running

factor10

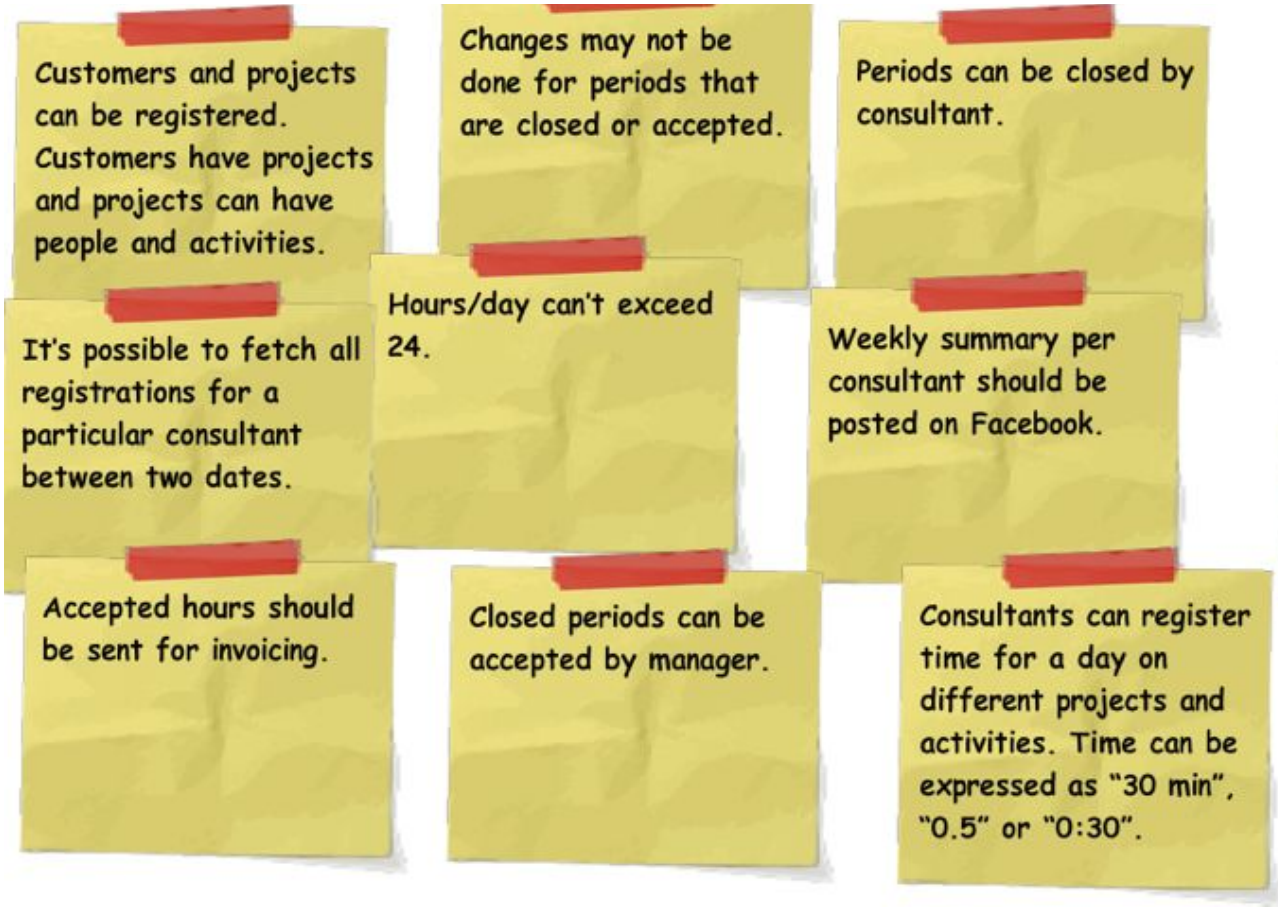
Lab 1: up and running

- → Git clone this repo: https://github.com/factor10/ddd-e2e-workshop
- → npm install
- → npm run test
  - → ... to run unit tests
- → npm run cypress
  - → ... to run the Cypress tests
- → npm run cypress:open
  - → ... to start the Cypress Test Runner UI
- → Run the Cypress test from the Test Runner UI

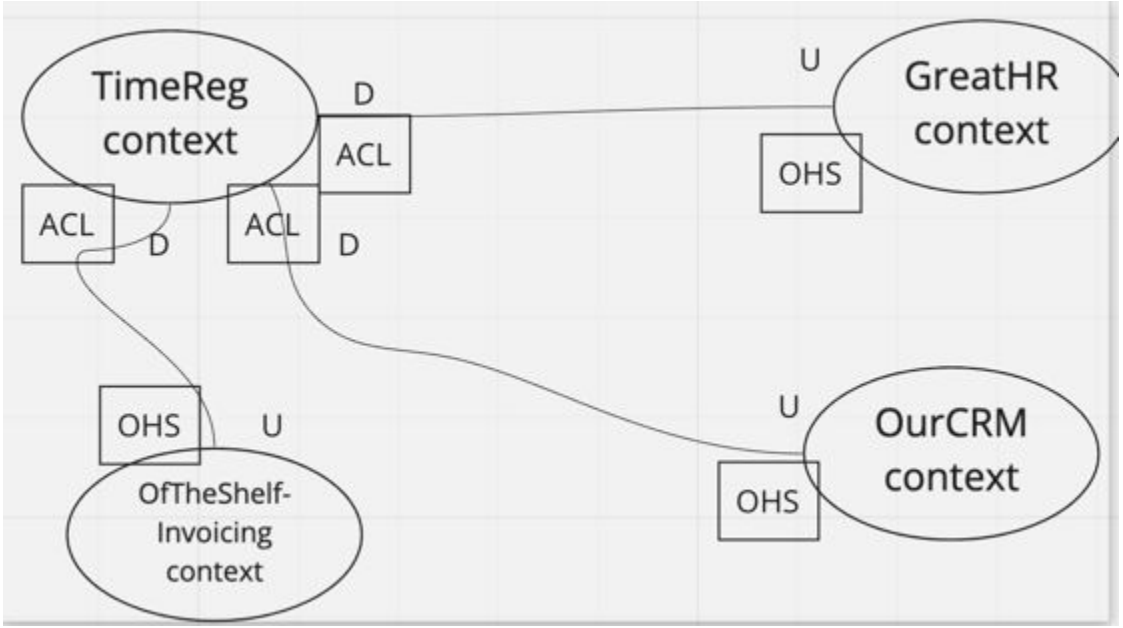
factor10

Intro to the domain

Lab app: Time reporting, requirements



Lab app: Context map

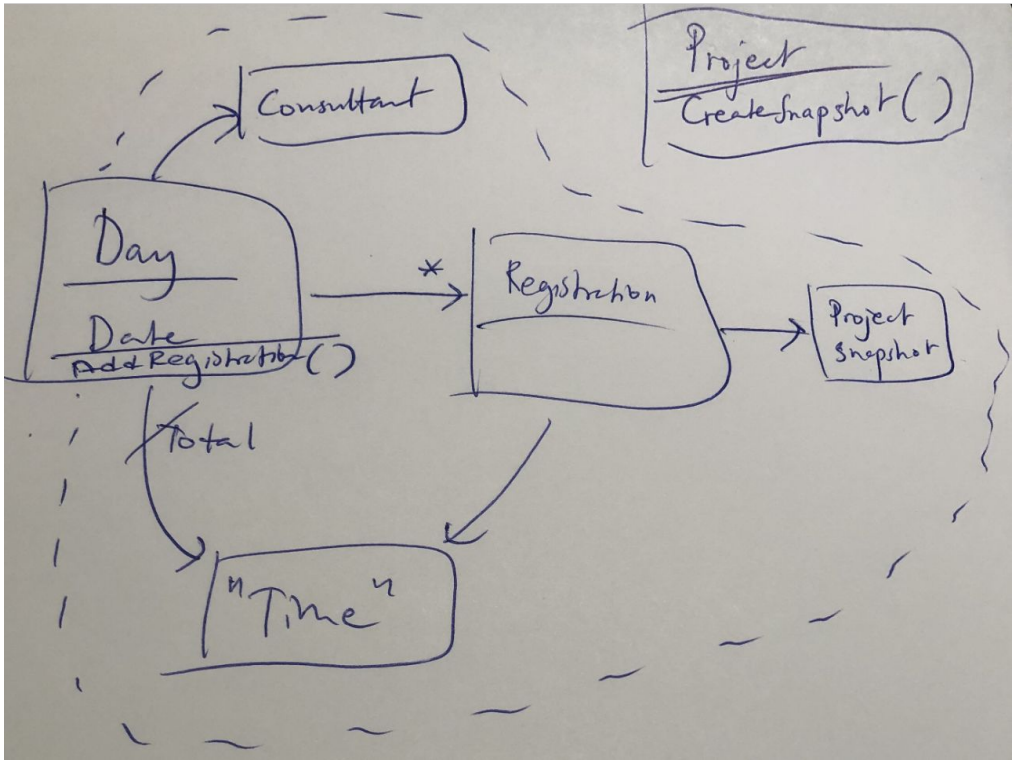


factor10

User story: Add time registration

As a consultant
I want to register my spent time
So that the customer is billed correctly

Lab app: Domain Model for time registration scenario



factor10

Intro to the code base

The HTTP API

Get all days in storage

```
GET  /api/days
```

Add registration to day for consultant

```
POST /api/days/{consultantId: Guid}/{date: yyyy-MM-dd}/registrations
```

Get all consultants

```
GET  /api/consultants
```

Get projects for consultant

```
GET  /api/consultants/{consultantId: Guid}/projects
```

factor10

Lab app: Folder structure

```
├─ src
│  ├─ anti-corruption-layer  # Anti-corruption layer towards other subsystems
│  ├─ api-controllers        # HTTP controllers to serve API
│  ├─ domain-model           # Domain model and infrastructure interfaces
│  ├─ env-config             # Config for each environment (development, test, e2e)
│  ├─ infrastructure         # Repositories
│  └─ public                 # Public static files for the web app
└─ test
   ├─ e2e                    # e2e (Cypress) tests
   └─ unit                   # Unit tests
```

factor10

Why bother with UI testing

factor10

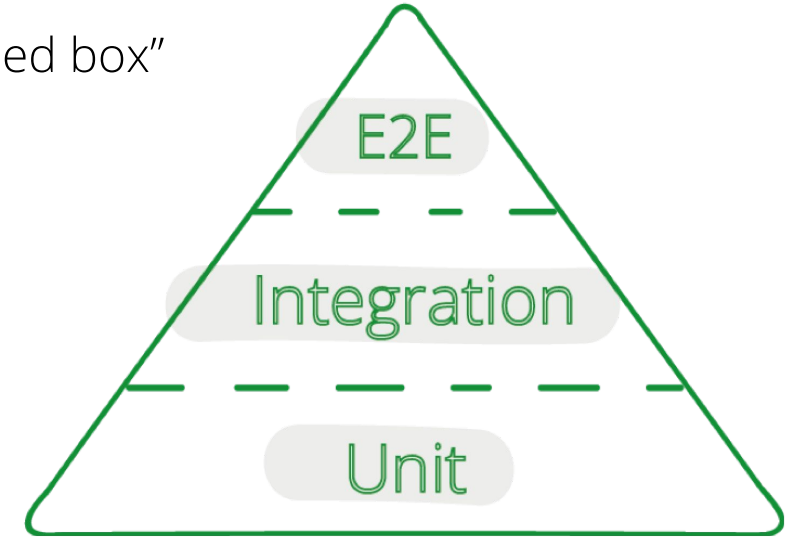
Old “truths”

- → UI tests fail most of the time, they are very brittle
- → UI tests are near impossible to troubleshoot
- → UI tests are difficult to create
- → UI tests are, just like manual testing, created after the production code
- → Setting up test data and state is way too hard

factor10

Why write UI test at all

- → Automated regression testing
- → Framework agnostic
- → Testing from a user perspective
- → Treats your entire app as a “closed box”



factor10

Syntax and tooling in Cypress

factor10

Why Cypress

- → Easy to get started
  - → `npm install` or with Docker image
- → Run test headless or with GUI
- → Full control of browser
- → GUI features:
  - → Auto reload of tests
  - → “Time travel” in test execution
  - → Access to DOM, network etc. via browser developer tools
- → Extendable framework
  - → Write your own helper functions and page objects
- → Automatic waiting and re-tries

factor10

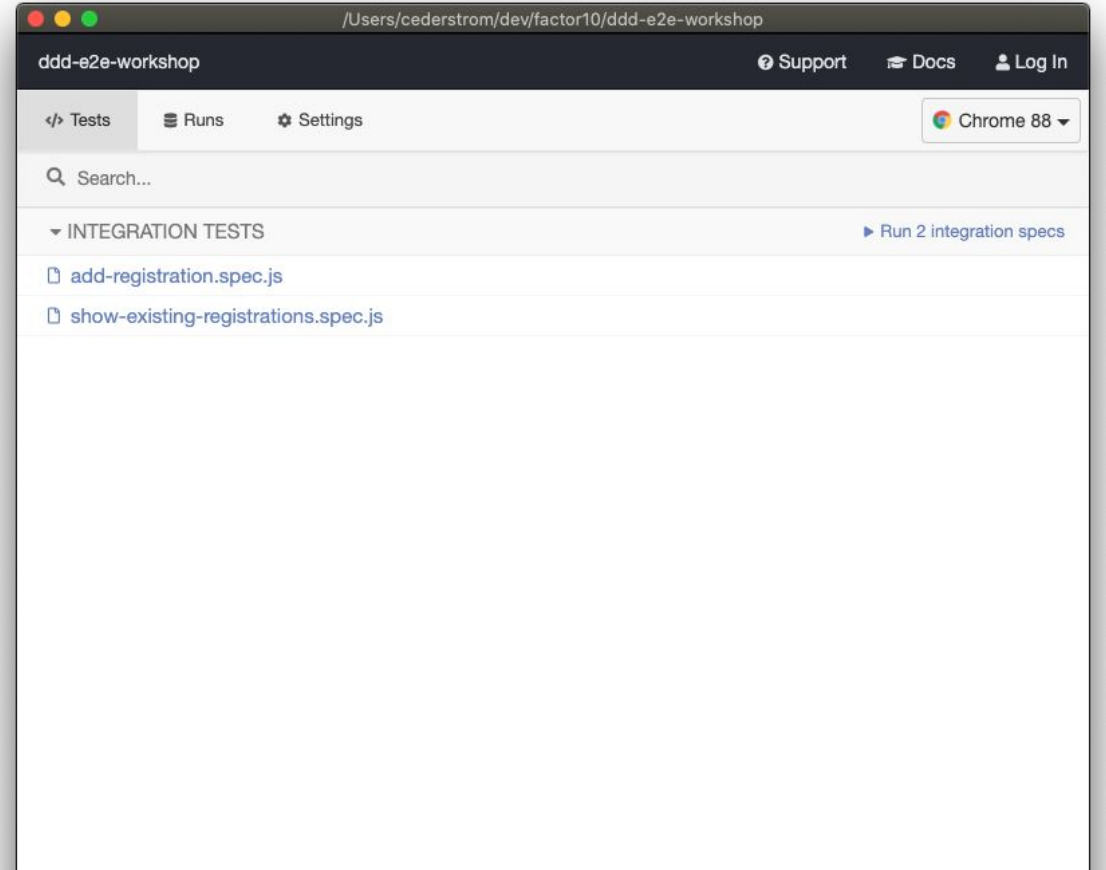
Automatic waiting

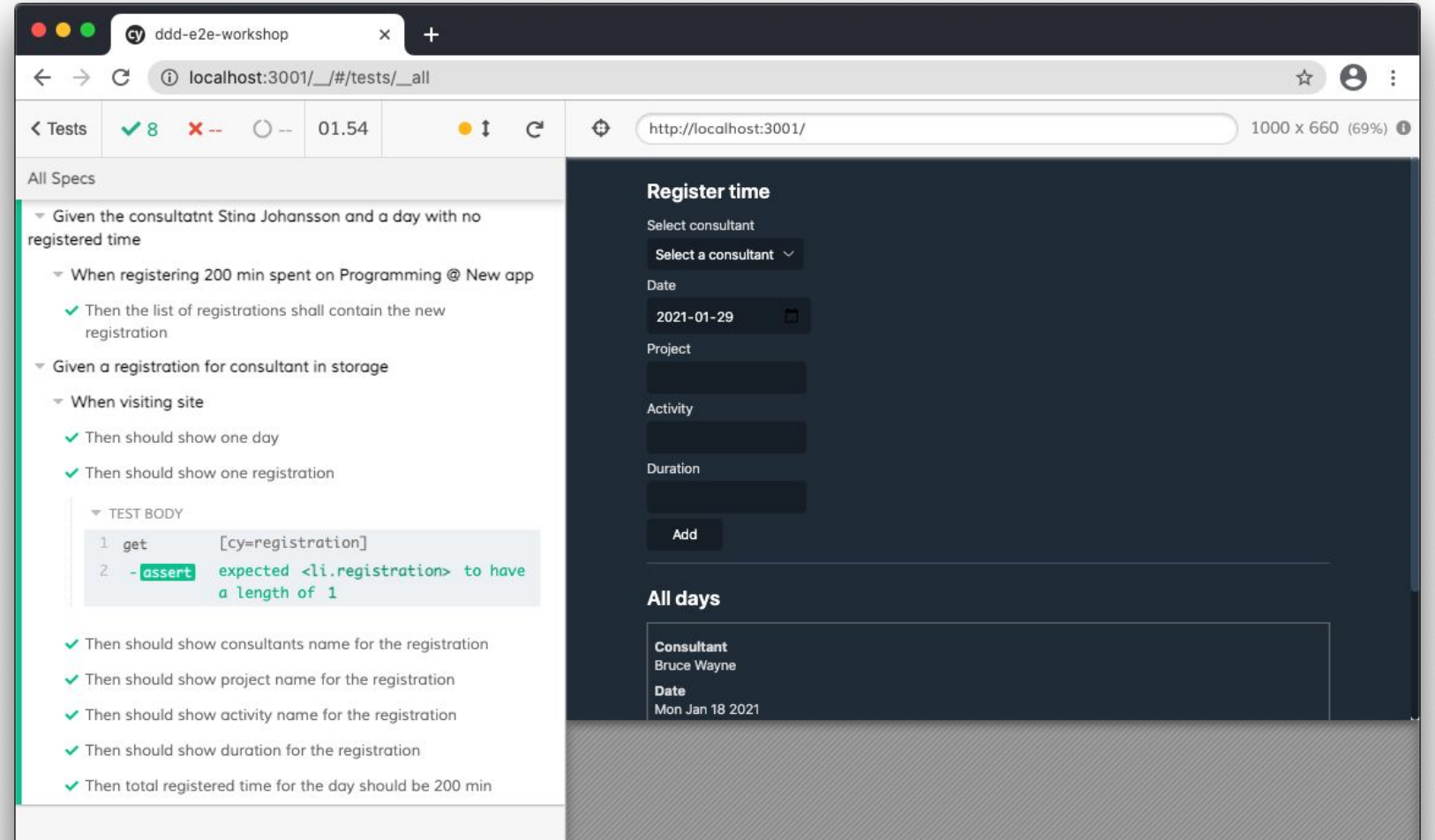
- → Waiting for...
  - → the DOM to load
  - → your framework to finish bootstrapping
  - → network request
  - → animations
- → Cypress wraps queries with robust waiting for you

factor10

```
context("Given the consultant Stina and a day with no registered time", () => {
  describe("When registering 200 min spent on Programming @ New app", () => {
    before(() => {
      cy.get("[cy=input-date]").type("2021-01-25");
      cy.get("[cy=select-project]").select("New app");
      cy.get("[cy=input-activity]").type("Programming");
      cy.get("[cy=input-duration]").type("200 min");
      cy.contains("button", "Add registration").click();
    });

    it("Then the list of registrations shall contain the new registration", () => {
      cy.get("[cy=registration]").should("have.length", 1);
      cy.get("[cy=consultant-name]").first().should("have.text", "Stina Johansson");
      cy.get("[cy=project-name]").first().should("have.text", "New app");
      cy.get("[cy=activity-name]").first().should("have.text", "Programming");
    });
  });
});
```


/Users/cederstrom/dev/factor10/ddd-e2e-workshop
ddd-e2e-workshop
Support
Docs
Log In
Tests
Runs
Settings
Chrome 88
Search...
INTEGRATION TESTS
Run 2 integration specs
add-registration.spec.js
show-existing-registrations.spec.js


ddd-e2e-workshop
localhost:3001/__/#/tests/__all
Tests
8
--
--
01.54
All Specs
Given the consultatnt Stina Johansson and a day with no registered time
When registering 200 min spent on Programming @ New app
Then the list of registrations shall contain the new registration
Given a registration for consultant in storage
When visiting site
Then should show one day
Then should show one registration
TEST BODY
1 get [cy=registration]
2 -assert expected <li.registration> to have a length of 1
Then should show consultants name for the registration
Then should show project name for the registration
Then should show activity name for the registration
Then should show duration for the registration
Then total registered time for the day should be 200 min
http://localhost:3001/
1000 x 660 (69%)
Register time
Select consultant
Select a consultant
Date
2021-01-29
Project
Activity
Duration
Add
All days
Consultant
Bruce Wayne
Date
Mon Jan 18 2021

factor10

Lab 2

test after

factor10

User story: Add time registration

As a consultant
I want to register my spent time
So that the customer is billed correctly

Scenario: Adding more time to a Day

Given that consultant Stina has registered 60 min Programming
on New app for 2021-02-02
When Stina registers 30 min Meeting on New app for 2021-02-02
Then Stina has a total of registered time of 90 min on 2021-02-02

factor10

Ubiquitous Language

To create a supple, knowledge-rich design calls for a versatile, shared team language, and a lively experimentation with language that seldom happens on software projects.

[Evans]

factor10

User story: Add time registration

As a consultant
I want to register my spent time
So that the customer is billed correctly

Scenario: Adding more time to a Day

Given that consultant Stina has registered 60 min Programming on New app for 2021-02-02
When Stina registers 30 min Meeting on New app for 2021-02-02
Then Stina has a total of registered time of 90 min for 2021-02-02

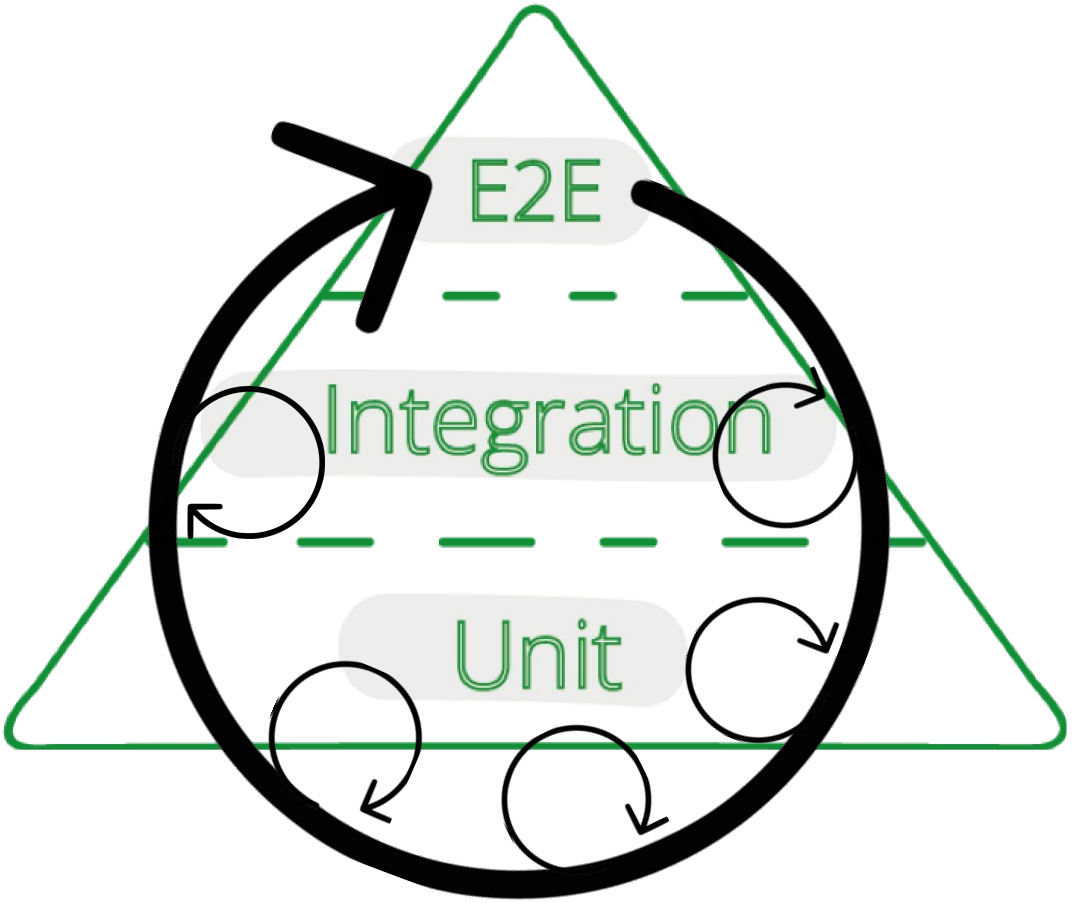
factor10

e2e UI test first

factor10

Challenging the old “truths”

- → UI tests fail most of the time, they are very brittle
  - → How often do they execute?
  - → Do you have full control of the state?
  - → Are they too large?
- → UI tests are near impossible to troubleshoot
  - → This is a tooling issue
- → UI tests are difficult to create
  - → Mostly tooling but also due to design
- → UI tests are, just like manual testing, created after the production code
  - → Why? Does it have to be like that?
- → Setting up test data and state is way too hard
  - → Is a system is test driven, will it not be more testable?


E2E
Integration
Unit

factor10

```
context("Given consultant Bruce has registered 200 min Programming @ New app", () => {
  before(() => {
    cy.get("[cy=select-consultant]").select("Bruce Wayne");
    cy.get("[cy=input-date]").type("2021-01-25");
    cy.get("[cy=select-project]").select("New app");
    cy.get("[cy=input-activity]").type("Programming");
    cy.get("[cy=input-duration]").type("200 min");
    cy.contains("button", "Add registration").click();
  });

  describe("When visiting site", () => {
    before(() => cy.visit("/"));

    it("Then should show one registration", () => {
      cy.get("[cy=registration]").should("have.length", 1);
    });
  });
});
```

factor10

```
context("Given consultant Bruce has registered 200 min Programming @ New app", () => {
  before(() => {
    // Get fixture `test/e2e/fixtures/one-registration.json` and overwrite storage
    cy.overwriteStorageWithFixture("one-registration");
  });

  describe("When visiting site", () => {
    before(() => cy.visit("/"));

    it("Then should show one registration", () => {
      cy.get("[cy=registration]").should("have.length", 1);
    });
  });
});
```

factor10

Lab 3
test first

factor10

User story: Add time registration

As a consultant
I want to register my spent time
So that the customer is billed correctly

New scenario:
✨Human readable total duration✨

Given that consultant Stina exists and has no registrations for 2021-02-02
When Stina registers 200 min for 2021-02-02
Then Stinas total duration for 2021-02-02 should be shown as 3 h 20 min

factor10

User story: Add time registration

As a consultant
I want to register my spent time
So that the customer is billed correctly

New scenario:
Filter list of days by consultant

Given Sina and Per has one registration each for 2021-02-02
When selecting Stina
Then only Stinas registration is listed

factor10

User story: Add time registration

As a consultant
I want to register my spent time
So that the customer is billed correctly

New scenario:
Filter list of days by date

Make BDD text executable

- → Try it if you have time in the lab…
  - → https://github.com/TheBrainFamily/cypress-cucumber-preprocessor

factor10

Now what?

factor10

Now what

- → Try to add e2e UI test for your next feature or fix
  - → Start small and iterate
- → The *first* tests will be difficult and time consuming to write
  - → But very soon they will add great value during development
- → Setting up state is the most difficult part
  - → By dealing with this first you will avoid an unmanageable situation

factor10

Good practices for e2e UI tests

- → Write tests from your *users' perspective*
- → Use your ubiquitous language and collaborate with the domain experts
- → Make the test runnable on *any* development machine first
- → Use a separate, production like, environment for e2e test
- → If you need data, use data dumps from production
  - → Make sure to obfuscate or "wash" sensitive information first!
- → If you can't control a part of your system: *stub it, fake it, mock it*

factor10

Good luck!

Andreas Cederström
andreas.cederstrom@factor10.com

Jimmy Nilsson
jimmy.nilsson@factor10.com