

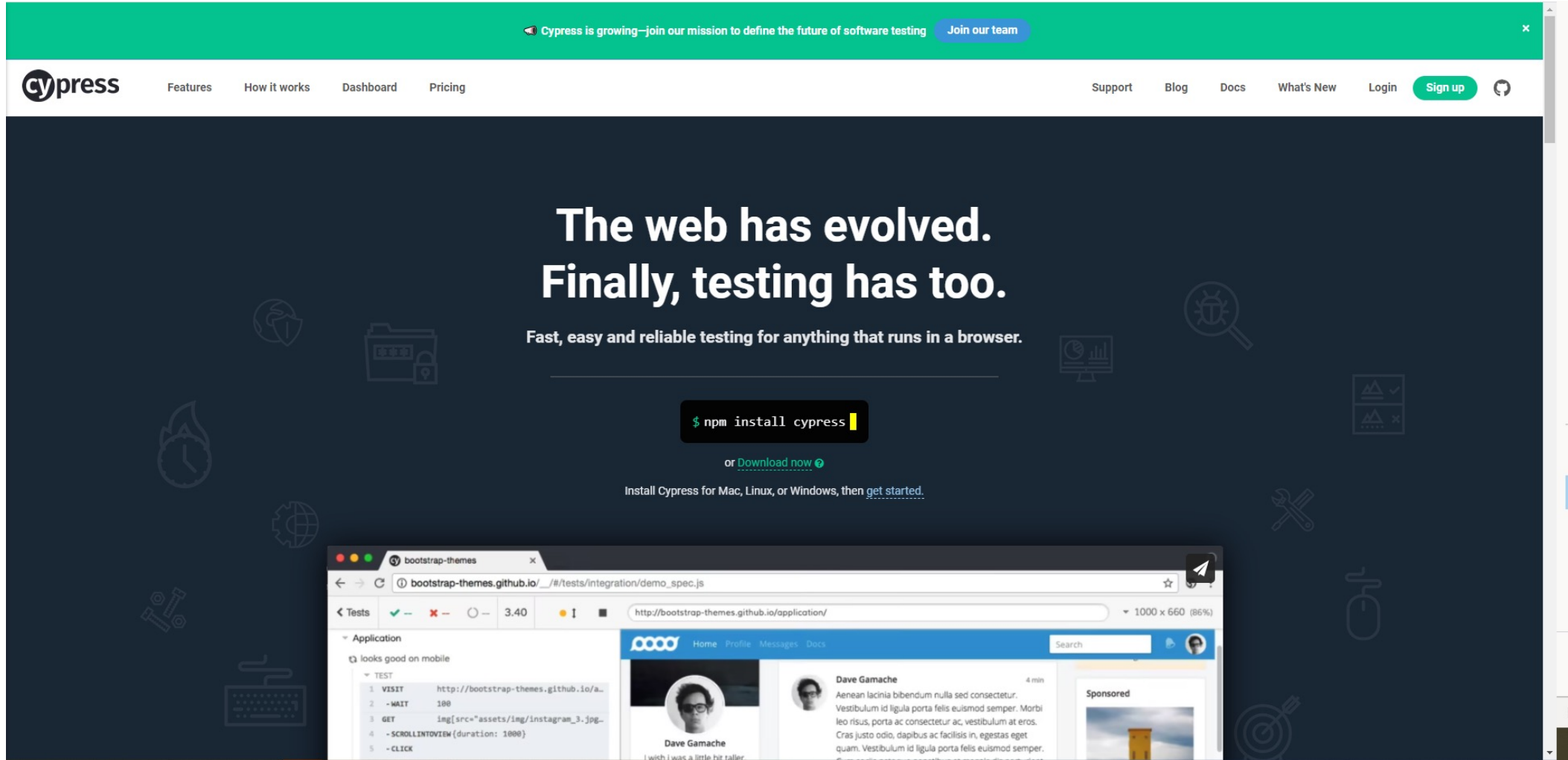
d-wise™

An Instem Company



Automated Testing of the Programming Workflow using Cypress

What is Cypress (cypress.io)?



The screenshot shows the Cypress.io website homepage. At the top, a green banner reads "Cypress is growing—join our mission to define the future of software testing" with a "Join our team" button. Below this is a navigation bar with links: Features, How it works, Dashboard, Pricing, Support, Blog, Docs, What's New, Login, and a "Sign up" button. The main content area has a dark background with the headline "The web has evolved. Finally, testing has too." and the subtext "Fast, easy and reliable testing for anything that runs in a browser." Below this is a terminal snippet showing the command `$ npm install cypress` and a "Download now" link. Further down, it says "Install Cypress for Mac, Linux, or Windows, then [get started](#)." At the bottom, there is a screenshot of a web browser showing a test suite for "bootstrap-themes" with a list of test steps: VISIT, WAIT, GET, SCROLLINTOVIEW, and CLICK. The browser window also displays a social media profile for Dave Gamache.

Cypress is growing—join our mission to define the future of software testing [Join our team](#)

cypress Features How it works Dashboard Pricing Support Blog Docs What's New Login [Sign up](#)

The web has evolved. Finally, testing has too.

Fast, easy and reliable testing for anything that runs in a browser.

```
$ npm install cypress
```

or [Download now](#)

Install Cypress for Mac, Linux, or Windows, then [get started](#).

bootstrap-themes
bootstrap-themes.github.io/_#/tests/integration/demo_spec.js
Tests 3.40
Application
looks good on mobile
TEST
1 VISIT http://bootstrap-themes.github.io/a...
2 -WAIT 100
3 GET img[src="assets/img/instagram_3.jpg-...
4 -SCROLLINTOVIEW {duration: 1000}
5 -CLICK

Home Profile Messages Docs
Dave Gamache
Aenean lacinia bibendum nulla sed consectetur. Vestibulum id ligula porta felis euismod semper. Morbi leo risus, porta ac consectetur ac, vestibulum at eros. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper.

Rationale for end-to-end test development

- Programmatically execute complex user workflows through the system
- Broad code coverage
- Enables identification of system regressions early in the development life-cycle not easily detected with unit or integration tests
- Gives agile development teams greater confidence to act aggressively in pursuit of goals and deadlines

d-wise strategy for end-to-end testing

- Identify critical workflows through the application
- Develop end-to-end tests to programmatically execute the critical workflows
- Execute tests at a routine frequency
- Integrate test reporting into key communication channels to ensure team members are informed of test execution results
- Act with urgency to identify and address root causes of test failures
- Maximize test credibility by addressing sources of false positives

Programmatic testing of the RBQR workflow

- RBQR – Risk Based Quality Review
- Exercises critical flow of a program through the web application
- Defines planned items
- Imports program dependencies
- Executes SAS program
- Evaluates program outputs

Preview

- Review video recording of actual test execution
- Highlight some Cypress tips & tricks to improve speed of test execution
- Demonstrate the power of Cypress to develop tests of highly complex user workflows

Challenging dogma

- Cypress breaks down end-to-end testing paradigms
- Stop using Page Objects and Start using App Actions:
<https://www.cypress.io/blog/2019/01/03/stop-using-page-objects-and-start-using-app-actions/>

```
if (w.Cypress) {  
  w.__store__ = store;  
}
```

Establish test preconditions

- BEFORE ALL hook establishes test pre-conditions
- Authenticate as an administrator
- Programmatically establish conditions for creating a new project tree
- Re-authenticate as a user
- Programmatically create new project tree
- Checkout the newly created project
- Executing tests in isolation is not always feasible in a complex user workflow

Test programming plan item CRUD

- CRUD – Create/Read/Update/Delete
- Planned items are the expected output from a program execution
- Visit the newly created project leaf (Reporting Effort)
- Exercise Create, Read, Update and Delete of planned items
- Exercise alternative means of creating planned items via spreadsheet import
- Exercise planned item copy/paste functionality
- With basic planned item functionality verified we are ready to exercise the program workflow in earnest

Upload program file and dependencies

- Upload program file fixture
- Upload fixtures upon which program execution depends
- Copy dependencies to appropriate locations
- Verify our newly introduced input files are in the proper locations for program execution

Development environment program execution

- Register the program within user's development environment
- Execute the program
- Monitor the user interface to verify successful program execution
- Verify expected program execution outputs in development environment

Pre-production program execution

- Promote program to pre-production location
- Run the program
- Expect special notification with information regarding presence of planned items
- Verify expected program execution outputs in pre-production environment
- Programmatically introduce the planned items necessary for program promotion
- Re-execute our program to create necessary conditions for promotion

Production program execution

- With pre-conditions satisfied, promote program to production
- Execute the program
- Verify state of programming plan items
- Verify program run details

Summary

- Developing enterprise applications for the pharmaceutical industry requires rigorous testing. Cypress is a next generation, open source, front end testing tool built for the modern web. d-wise has adopted the Cypress framework to speed the development of rigorous and comprehensive software tests.
- The RBQR workflow is a complex user workflow. Leveraging Cypress we've succeeded in developing e2e tests that programmatically execute the workflow in less than 4 minutes.
- We hope this presentation will aid others in our industry in the development of quality solutions for the pharmaceutical industry.

Contact

Darwin Hadley

Senior Software Engineer

Darwin.Hadley@d-wise.com