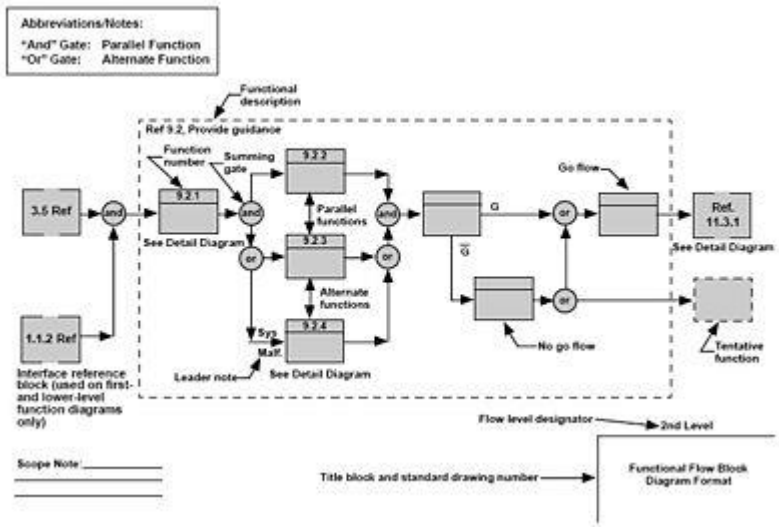
Determine The Requirements (Customer Journey Maps)

Being able to recognize the telltale signs of customer satisfaction or dissatisfaction helps businesses proactively launch personalized campaigns to either keep users engaged and prevent them from churning.

One way to do this is by identifying the interactions that increase customer value and loyalty. For instance, at Segment we studied the behavior of our loyal customers and identified a set of high-value actions they all had in common. Using this data, we crafted personalized messages and engagement campaigns to nudge *other* customers to follow suit.

Requirement Analysis (Functional, Operational, Technical) / Flow Charts

A **functional flow block diagram** (**FFBD**) is a multi-tier, time-sequenced, step-by-step flow diagram of a system's functional flow.[2] The term "functional" in this context is different from its use in functional programming or in mathematics, where pairing "functional" with "flow" would be ambiguous. Here, "functional flow" pertains to the sequencing of operations, with "flow" arrows expressing dependence on the success of prior operations. FFBDs may also express input and output data dependencies between functional blocks, as shown in figures below, but FFBDs primarily focus on sequencing.



Technical Architecture

In a fast-paced digital world where new technical innovations flood the market every year, businesses need to act quickly to keep up with the competition. However, adopting random technologies without knowing how they will create more value, in the long run, can have detrimental effects on a business. In addition, cloud computing, microservices, and distributed systems are adding another level of complexity to the IT landscape. All these factors combined have created a growing need for skilled IT architects.

IT architecture aims to align the business strategy with new technological solutions and consists of various disciplines ranging from strategic to highly technical. One that fits into the latter category is technology architecture as it mainly focuses on the design and documentation of software applications. Thus, technical architects create blueprint schematics of technical solutions making sure that new products or systems meet specified requirements.

Open Source Frameworks

This year's report reveals that the top three open source frameworks are the same as last year’s:

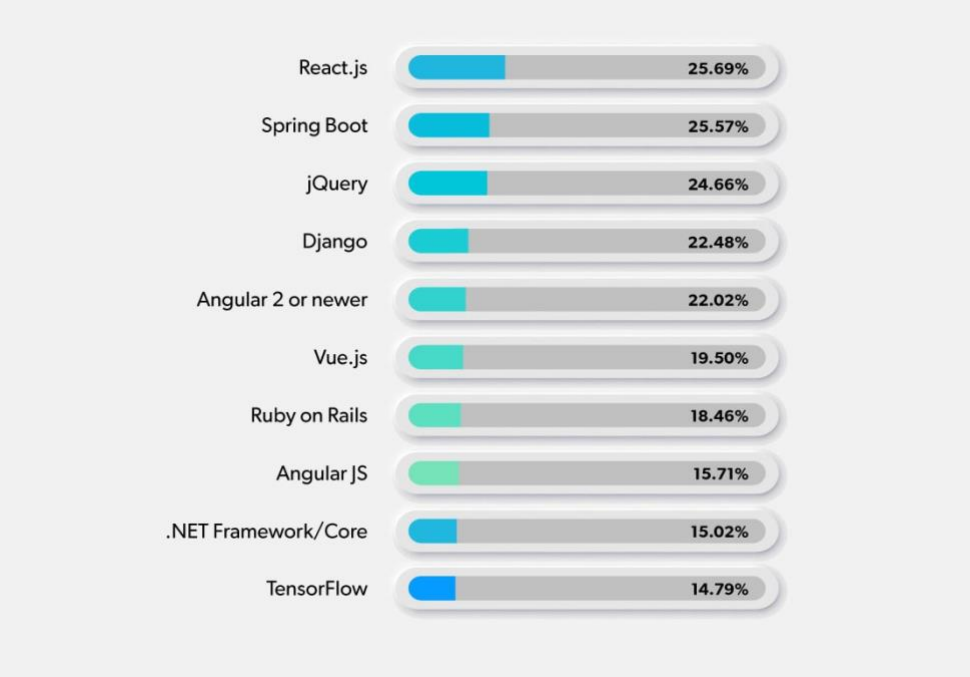
1. React.js

2. Spring Boot

3. jQuery

Usage of these three technologies is virtually identical, within one percentage point of each other. While jQuery is technically not a framework, we felt it made sense to include it here, as it is a very useful and complete JavaScript library. It's interesting to see a number of JavaScript frameworks being used across industries.

The graph below shows the top 10 open source frameworks, but in total we asked about 20 different frameworks. If you're curious about a framework you don't see here, refer to pgs. 33-34 in the report.



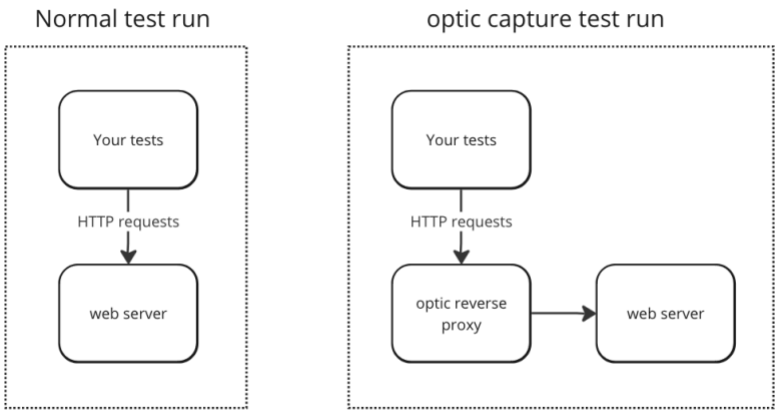
Other things to note: Usage of Django, the web framework for Python, is up by 6% compared to last year, and Angular and Vue.js (JavaScript frameworks), declined by 4% each.

Third-Party API's

This will generate an example configuration that you will need to edit for your tests and server. You will need to specify:

- A command to run your server (`server.command`)
  - you can omit this if the server is already running (e.g. against a remote server) or is started separately
- the URL where your server can be reached (`server.url`) (e.g. `http://localhost:3000` or `https://api.example.com`)
- A command to run your tests (`requests.run`)

- Your tests must make requests through the http layer (Optic captures network traffic and uses a reverse proxy to capture these requests)
- You will also need to update your test script to send requests to Optic's reverse proxy, this will be available in the environment variable OPTIC_PROXY



Cloud Deployment

With an effective cloud deployment model, an organization achieves numerous benefits, including:

- **Faster and simplified deployments.** Automate builds that deploy code, databases and application releases, including resource provisioning.
- **Cost savings.** Control costs using consumption-based pricing and eliminate capex-heavy on-premises environments.
- **Platform for growth.** Leverage the global infrastructure provided by cloud service providers (CSPs) to seamlessly expand the business into other geographies.
- **New digital business models.** Exploit the continuous release of features and services by CSPs, incubate new technologies and innovate digital business models.
- **Business resiliency.** Architect for the availability and fault-tolerance CSPs offer and ensure disaster recovery and business continuity of applications to make the business resilient.
- **Agility and scalability.** Use autoscaling and scalability to meet peak demands of the business without provisioning for excess capacity.
- **Geographic reach.** Access applications from any location, on any device, leveraging the connectivity backbone of CSPs.