PDFTRON

Changing the way the world works with documents.

6 Important Factors when Choosing a PDF Library

Adam Pez

A PDF DOCUMENT

Overview

You plan to embed PDF functionality into an application. But before you dive into the project, you must decide: do you go with a more expensive commercial PDF SDK — or a lower-cost alternative such as an open-source library or an open-source wrapper?

There are non-trivial costs to switching later. Developers have to re-learn the new library, re-adjust the backend, customize the UI to match what users are accustomed to, as well as migrate documents, form data, annotations, and more.

According to market research conducted by Stax Inc., the average Net Promoter Score (NPS) for the top five PDF SDK vendors is 35%. And 70% of customers express interest in switching despite the high costs.

This dissatisfaction implies that picking the right PDF SDK is a lot harder than it seems. And to help you avoid the same mistakes as past implementations, we've written this article.

(We also recently surveyed 57 unique organizations that switched from PDF.js to a commercial SDK. Read our comprehensive guide to PDF.js to learn more.)

Restrictions on Features and Platforms

A first mistake organizations make when selecting a PDF library for the first time is to assume fixed feature requirements. But these are likely to evolve.

Users start to ask for more functionality as they grow dependent upon a PDF SDK. An organization will then have to consider saying no to user feature requests; building time-intensive and challenging customizations on top; or integrating additional libraries and thus adding more complexity, maintenance overhead, and risk. Additionally, a library may work fine initially on the main platforms preferred by your users. But later if you wish to expand, the library does not support the platforms you want — or the APIs are inconsistent, with different classes and methods across platforms making it so your engineers have to start from scratch.

To avoid this hidden cost, go with an SDK with a broad feature-set across multiple platforms, providing you the flexibility to grow down the road.

> Maybe big companies can absorb the costs of maintaining three-to-four different relationships with different vendors, each with a different code base, different roadmaps, and different problems. I'm not saying it isn't possible.
>
> *Kalsefer Co-Founder and CEO, Avshi Segev*