If you want the **TL;DR** version, you can skip to the summary and recommendations.

History

According to the git history, Jakob Truelsen started this project the day after Qt 4.4 introduced QtWebKit. The initial commit seems more a demo, but soon grew to have a lot of new features. Although Qt 4.5 enhanced the QtWebKit API, the focus wasn't on the HTML to PDF conversion use case, so the first Qt patch got introduced. As an example of how far ahead of the curve this was, the equivalent support was added for Chrome 64 in 2017 – more than 8 years later!

During the early years, the focus of both Google and Apple was the web platform – printing (*especially to PDF*) simply wasn't a focus area. So although there were attempts to upstream the API changes, there wasn't much interest in getting them merged. At the same time, the only other capable tool at that time (*which is still going strong*) was PrinceXML, but it was commercial and had steep license fees so the popularity of wkhtmltopdf kept on exploding.

> Fun fact: Ariya Hidayat, who did an initial review of the API changes was then working at Nokia (*which had acquired Qt in 2008*) and would go on to develop PhantomJS, which too was based on QtWebKit and enabled a whole class of other tools to be developed.

Even as the upstreaming work stalled, Jakob and other contributors kept developing new features and extending the Qt patches. Once Qt 5.0 was released in 2012, due to a quirk in the way C++ libraries are developed meant that no new APIs could be introduced in the Qt 5.x lifecycle.

During the 2012-2014 timeframe, the project was stagnant – the number of users kept on increasing but the number of developers didn't. This resulted in a lot of issues being created in the tracker. In most open-source projects, some of the users eventually step up and become contributors – which didn't happen as the project was an intersection between Qt and WebKit (*both written in C++*) while most users were just familiar with HTML, CSS and JS.

Meanwhile, interesting things were happening in the wider world. Google decided to fork WebKit into Blink in April 2013, and Qt decided to follow with it by announcing Qt WebEngine within 6 months. Although their plan was to keep developing both, it didn't work out and QtWebKit was eventually deprecated in 2015 and removed in 2016. It had been on life support for years ever since it was removed from WebKit within a month of the Qt WebEngine announcement.

Back in wkhtmltopdf-land, a shiny new 0.12.0 release happened in early 2014. Ashish Kulkarni became the maintainer after that and all further releases in the 0.12.x series were made by him (*in case it isn't obvious, he's the author of this long **and** boring essay* □). However, things weren't rosy: the 0.12.0 release was based on Qt 4.8.5 but Qt 5.0 had been already released just under 2 years ago – so there was an urgent need to update the underlying engine.

Initially, QtWebKit 2.3 seemed promising but it turned out to be a red herring: it was supported mostly by Linux distributions and not officially by Qt. Although there were some contributions to both Qt4 and Qt5Base/Qt5WebKit in this time period – it was a small fraction of the patches that needed to be upstreamed. Less than a year later in 2015, QtWebKit would be deprecated and there was effectively **no upstream to contribute to**.

History tends to repeat itself, and the same thing happened in 2015-2016: too many users, too few developers and not enough clarity on what was to be done to take things forward. At the same time, there was growing awareness about the sorry state of

WebKit security (*for those who read it, wkhtmltopdf depends on the WebKit1 in-process API*). It looked like things would improve with a revamped QtWebKit fork which would be accepted by most Linux distributions, but that turned out to be a mirage.

Regardless, an initial plan for 0.13 major release was drawn and work started. But work on the revamped QtWebKit stalled after 5.212 alpha2 in 2017 and so did the motivation to continue on 0.13 . Although the revamped QtWebKit has revived again in 2019, received funding via Patreon/GitHub and annulen has made great progress, the status shown by GitHub (*as of 2020-06-10*) shows how much it has to catch up:

> This branch is 2947 commits ahead, **9266** commits behind WebKit:master.

Plus Qt 6.0 is going to ship this November, and there's uncertainity over the future of Qt itself. Meanwhile, Chrome has made great strides since 2016 when they started making Printing/PDF a focus – this same fact led to a significant maintainer stepping down for PhantomJS. Also, if you see the puppeteer page.pdf API, it looks eerily similar to the options used by wkhtmltopdf – which is a good thing, as it's well supported and has a much more modern browser engine

But this has led to a Blink monoculture – almost everyone uses the same browser engine, which doesn't feel healthy for someone who lived through the IE 6 days, I personally had to support that monstrosity even as late as 2014 (*in a very conservative banking context*). Even though Google is unlikely to do that, things can change in the future so more competition is always good. *Note that these are the personal views of Ashish Kulkarni, not of anyone else.*

Summary

- Qt 4 (*which wkhtmltopdf uses*) hasn't been supported since 2015, the WebKit in it hasn't been updated since 2012.
- Qt 5 is supported, but removed QtWebkit in 2016 (Qt 5.6), development stopped after 2012 but minor fixes continued till 2015.
- QtWebKit 5.212 by annulen uses a version more than 4 years old, but is packaged by major Linux distributions.
- qtwebkit-dev-wip uses a version of WebKit almost 1.5 years old, and isn't ready for release yet – packaging by distributions comes later!

Where do you contribute to upstream the patches? It makes sense to only do the effort if it's going to be maintained – browsers have a fast release candence exactly for this, to address security issues. If you wish to donate money, please sponsor QtWebKit instead … that'll help more projects than just this one and will ensure that there **is** a future.

Future Plans

- After the 0.12.6 release, I'll do a final 0.12.7 release by Aug 2020 which fixes any 0.12.6 regressions and review/merge already submitted PRs by a lot of people (*ignored till nowdue to personal reasons*)
- Work on rebaselining the patches to QtWebKit 5.212 – although outdated, it'll be a practice run to see if it's possible at all.
- If the above point is successful, submit them to the Qt and QtWebKit projects and get them merged after review, changing wkhtmltopdf as required.

There is no deadline by when the last two points will happen, or even that they will be done at all – it all depends on the time available to the maintainer and if volunteers step up to take up some tasks. So, please stop asking about that

Recommendations

- **Do not use wkhtmltopdf with any untrusted HTML** – be sure to sanitize any user-supplied HTML/JS, otherwise it can lead to complete takeover of the server it is running on! Please consider using a Mandatory Access Control system like AppArmor or SELinux, see recommended AppArmor policy.
- If you're using it for report generation (i.e. with HTML you control), also consider using WeasyPrint or the commercial tool Prince – note that I'm not affiliated with either project, and do your diligence.
- If you're using it to convert a site which uses dynamic JS, consider using puppeteer or one of the many wrappers it has.