

SignalR – Stakeholder analysis

Daniel Saxén

38768

dasaxen@abo.fi

Table of contents

Stakeholder analysis	1
Acquirers.....	1
Assessors	1
Communicators.....	1
Developers.....	2
Suppliers.....	2
Support staff.....	2
Testers.....	3
Users	3
Competitors.....	3
Power/Interest grid.....	4
Conclusions	6
References.....	6

Stakeholder analysis

In this report section, the different stakeholders of the SignalR library [1] are analyzed. First, the different stakeholder classes for SignalR are classified. Then, a power/interest grid is used to prioritize the stakeholders. Finally, a short conclusion is presented.

Acquirers

If you view acquirers as the sponsors or investors of the project, Microsoft has developed the ASP.NET Core infrastructure together with the community. The development of SignalR is in Microsoft's interests, but SignalR is not really one of their primary focuses. Microsoft incorporated the .NET Foundation in 2014 to support the software development of the .NET Framework, including SignalR.

Financially speaking, the .NET Foundation can help to process private donations [3] and ensure that the funds are spent appropriately, but they do not pay for developers. Since the SignalR library is open source but designed for ASP.NET Core, the top contributors are working for Microsoft, in the ASP.NET team.

Assessors

The ASP.NET team is ensuring the system's conformance to standards by checking the contributed Github code thoroughly before adding someone's pull request to the development branch of the library.

Communicators

Communicators are the people who "explain the system to other stakeholders" [2]. Microsoft has documented quite a lot about ASP.NET Core, for example at [4]. There is an introductory presentation at [5] regarding ASP.NET Core SignalR. We assume that the top contributors of the SignalR library also are communicators, since they work for the ASP.NET team.

Developers

Currently, the ASP.NET Core SignalR library has 35 contributors on Github [6], the most frequent being [moozzyk](#) and [davidfowl](#). Both are working for Microsoft in the ASP.NET team. We assume the ASP.NET team members working on the SignalR library are the leading developers.

SignalR is an open source library, which means anyone with a reasonable proposition can contribute to the library. If you want to contribute, you must sign a Contributor License Agreement using your Github credentials. The CLA is found [here](#). In the agreement you must state that you are the sole owner of your contribution and that you are not making the contribution during working time elsewhere. After you have signed the CLA, you can submit the pull request, which still needs to be approved by the ASP.NET team.

Suppliers

Microsoft developers provided the initial infrastructure for ASP.NET Core. Microsoft also provides the .NET Core SDK and the Visual Studio IDE. Of course, any editor can be used, but Visual Studio is the most used IDE with ASP.NET. Visual Studio is often used for ASP.NET Core because it has comprehensive support for developing, testing and debugging web applications and C# as a built-in language. SignalR uses Github for version control and source code management. The library developers use a client, which is written in TypeScript. The .NET foundation can negotiate

Support staff

Open source projects in alpha phase, like SignalR, usually do not have any dedicated help desk or customer service departments. The .NET Foundation can help in various issues. On the .NET Foundation site, they state that “The .NET Foundation also provides administration and support for a number of .NET open source projects assigned to the foundation.” You can also get support by the ASP.NET team, for example by creating an issue at [7].

Testers

The contributors need to provide tests for every completed bigger feature [8]. The ASP.NET Team specifies “If there is a scenario that is far too hard to test there does not need to be a test for it. ‘Too hard’ is determined by the team as a whole.” Visual Studio can be used for unit tests.

Before accepting a GitHub contribution, an ASP.NET team member checks the contribution thoroughly, even if you provided necessary tests. The problem with having the ASP.NET team developers as the main testers of SignalR is that they know the system too well. Having people with less knowledge about the system test it would make the testing more objective [2], which would make the code more understandable to new contributors.

The ASP.NET team uses two continuous integration services for SignalR. They are called AppVeyor and Travis CI, which are used for building and testing the SignalR development branch after every commit. Github projects like SignalR can easily be synced with these CI services.

Users

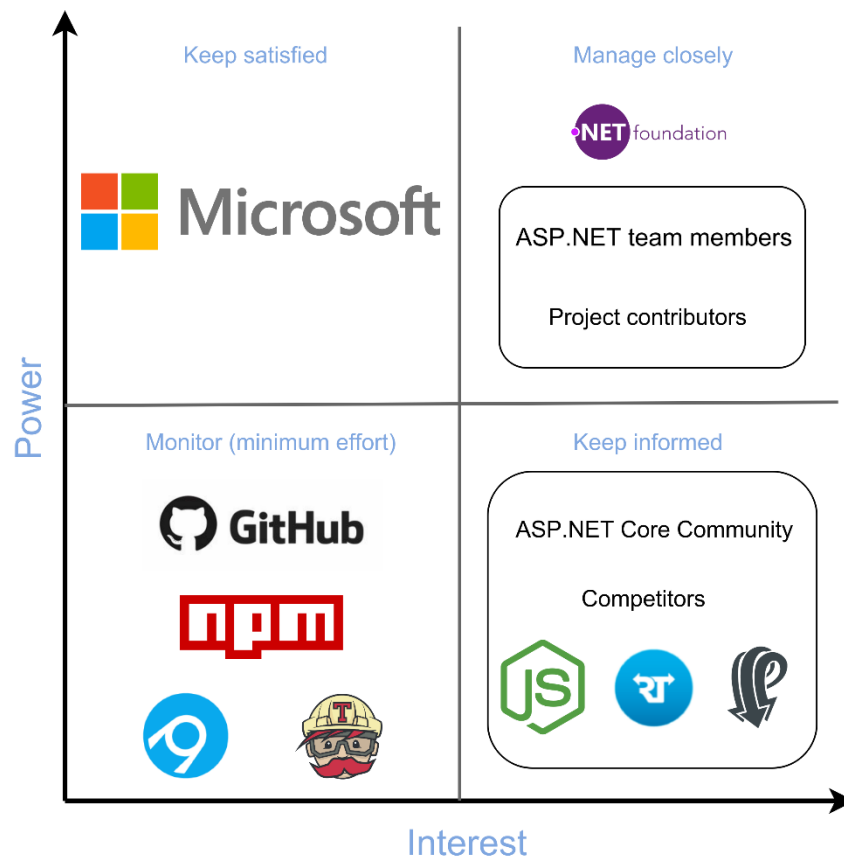
SignalR is under the Apache license 2.0, granting any user the rights [9] to freely use, modify and distribute software based on SignalR. The users of the system are the web developers using ASP.NET Core as their development framework. The users can also become developers by contributing to the SignalR Github repository.

Competitors

The competitors to SignalR are motivated to know how the SignalR development advances, so that they can use the information to plan their own projects. Node.js, Realtime Framework and Pusher are briefly discussed in the context view analysis. The difference between SignalR and the competitors is that the competitors have software which is far more used than SignalR, whereas SignalR is mostly used by a limited amount of developers (the .NET Core web developers).

Power/Interest grid

The power/interest grid below is a tool used to decide which of the stakeholders should be prioritized. The classical power/interest grid divides the stakeholders in to four different quadrants, which are presented and discussed below.



- **Low interest, low power:** This is the least important category of stakeholders. These stakeholders do not require much attention. In the lower-left quadrant is the web-based version control repository called Github, the JavaScript package manager called npm and the two continuous integration services AppVeyor and Travis CI. These stakeholders need minimum effort because they can easily be replaced by competitors (for example, you can use Bitbucket instead of Github for version control).

- **Low interest, high power:** Microsoft needs to be satisfied but they have many ongoing projects. SignalR is a relatively small library in a very early development stage which means that Microsoft is not highly interested in it. Microsoft is for the time being, unfortunately for SignalR, very interested in the booming cloud market with their latest cloud startup acquisition [10] two months ago.
- **High interest, low power:** The developers using ASP.NET Core do not have almost any power unless they choose to be project contributors on Github. They do have high interest, though, as the SignalR functionality would help the developers to implement real-time functions in their apps. The other group that has a high interest, but low power, are the competitors.
- **High interest, high power:** The most important group of stakeholders. You need to always keep these stakeholders informed and satisfied. The project contributors (apart from the ASP.NET team) have moderate power and high interest. They cannot make substantial changes to the SignalR library without the consent of the ASP.NET team, but they are still an important part of the project.

The ASP.NET team working on SignalR has high power and high interest. The team members are in many different stakeholder classes; developers, assessors, communicators, support staff and testers. This problem is briefly discussed in the concluding part of the stakeholder analysis.

The .NET Foundation supports the open source projects for the .NET platform by functioning as a forum for developers and by increasing the exposure of open source projects. For example, the .NET Foundation provides the automated CLA for Github projects and can provide Azure hosting for project resources.

Conclusions

The stakeholder group for SignalR is rather small. When you look at the group, you see that many of the stakeholders are somehow linked to Microsoft; The ASP.NET team works for Microsoft, the .NET Foundation was incorporated by Microsoft and ASP.NET Core, which this library primarily runs on, was developed by Microsoft and the community.

According to [2], a small group of stakeholders can either be good or bad. If you have a large group of stakeholders, you need to invest more resources to keep all their unique needs satisfied. Open source projects in alpha phase do not tend to have much personnel to manage a large stakeholder group.

On the other hand, having a small group of stakeholders has its setbacks. It is not a good thing from an architectural point of view that the ASP.NET team is in so many different stakeholder classes. For example, it would be recommendable to have someone outside the team to document the SignalR library, so that people could get a deeper understanding of the project. With better documentation, there is no doubt that there would be more contributors on Github.

References

1. <https://github.com/aspnet/SignalR>
2. https://moodle.abo.fi/pluginfile.php/300532/mod_resource/content/0/swarch-book.pdf
3. <https://dotnetfoundation.org/about> , fetched 01.10.2017
4. <https://docs.microsoft.com/en-us/aspnet/core/> , fetched 10.10.2017
5. <https://vimeo.com/204078084>
6. <https://github.com/aspnet/SignalR/graphs/contributors> , fetched 17.10.2017
7. <https://github.com/aspnet/SignalR/issues>
8. <https://github.com/aspnet/Home/blob/dev/CONTRIBUTING.md> , fetched 10.10.2017
9. <http://www.apache.org/licenses/LICENSE-2.0> , fetched 5.10.2017
10. <http://nordic.businessinsider.com/microsofts-acquires-cycle-computing-away-from-amazon-google-2017-8?r=US&IR=T> , fetched 12.10.2017