**ITCR SIP Kitware Discussion**

**11/13/2019**

**Stephen Aylward, Will Schroeder, Mike Grauer**

[More detail availabe at: https://docs.google.com/document/d/1OQb1DmL7ttihyLHwRkwDlwwoHjZOiaeNt2\_BBNj2bN0/edit](More%20detail%20availabe%20at:%20https://docs.google.com/document/d/1OQb1DmL7ttihyLHwRkwDlwwoHjZOiaeNt2_BBNj2bN0/edit)

Customers are asking about sustainability as well. Recently met with CZI and discussed s/w process and building community.

Almost all Kitware software is open source. Business is focused on custom application development and collaborative R&D with partners.

WS: What is the most important criteria evaluated by industry when evaluating OS softare? Refer to sustainability matrix (see Google Doc). Considerations include funding, solving an important problem, community dynamics. Does the community have a vital culture and passionate people?

CD: Can community outweigh some of the other factors?

WS: Yes, this can be overlooked and can be the most important factor. Governance, community, etc. Needs to be elevated.

SA: Goes both ways. Can have excellent funding but no community spirit, encouraging of contributions, etc.- these can degrade funding. A good community can overcome insufficient funding. This is often undervalued and not given the effort that it needs.

WS: Hiring good people is really hard so developing a community, training, materials is really important.

SA: Moving software from academia to industry, need professional support behind it. For companies to take on 3rd party software is a level of risk. OS used to have reputation of “wild west”. Offering professional support for OS software is comforting to small businesses.

CD: ITCR gives a lot of attention to user base. Can you elaborate on difference between community dynamics and user base?

SA: Criteria of a thriving community includes user base but more important is contributions back to the software from the users. How many contributions outside the core developers? Eg 60% of IDK contributed by “outsiders”. They had clear pathways to contribution and recognition. Then users have a vested interest in the software. This is a unique way to assess the success of an OS community.

WS: Small companies want to take advantage of software from an academic collaborator. Often we have commercial entities that want to use cutting edge features.

SA: When you move into final product that you want to move to FDA, adopting companies may do their own implementation. If this is key to the success of their product, will re-implement in their environment and integrate into their systems.

CD: Do you see any way for the commercialization to be a separate process or could it co-exist with a non-commercial version of the software? If there is a commercial interest, will research arm be abandoned>

WS: We are working with a simulation company that is incorporating one of our modules into our system. They want to contribute back to the OS software for maintainability. Leveraging the popularity and usability of the OS platform. They take little pieces and put it back into their systems.

SA: Commercial entities often find the bugs, etc, and harden the software. Needs to be much more robust than in an academic environment. Eg does the current registration technique outperform their current one? If it’s easy to swap in and try these different algorithms, that’s an advantage. Massive cost savings if this is easy to do. Some companies run both a research version of their code and a production version of their code.

For a company that doesn’t have a product and just want to do a rapid prototype >> this is the largest portion of their business. Don’t have to worry about the regulatory process at this phase, just want a demo that works. Aren’t going to be the big users of the toolkit. Harder to get something from them in terms of code contributions.

WS: A lot of vendors are scared of “lock-in”. Want freedom to do research.

GS: What is your business model?

WS: Build OS platforms and then work with others to provide 1) basic support 2) customization 3) collaborative R&D 4) custom applications.

SA: Key phrase is “open IT”. If you pay us to develop, you own it. Commercial company has the option of keeping it proprietary. Kitware doesn’t try to file patents, no copyright, no GPL. We give as much away as possible to build reputation.

WS: Some unexpected OS contributions, eg. Oil and gas companies! They understand this reduces maintenance burden, avoids vendor lock-in.

SA: Kitware is now 200 person company, worldwide. Thriving, profitable company.

CD: Do you prefer to be involved early in the development of a new technology?

SA: Hardest projects are when they have a large code base developed without the proper controls in place. Controls only work when they are seamless with the development process. Too disruptive to change. Use GitHub as an OS way of managing software. Use the review process inherent in GitHub and make this part of the culture. Starting with this gives you a solid base. Critical that it happens from the start. Also, very difficult to change the licensing. Almost no one switches from closed to open, GPL > Apache doesn’t happen. Important that it looks like all the code is written by the same person.

SA: Like the Apache2 because it covers patents and contributions. BSD is less clear. Contributors agree to apply the Apache2. Don’t need to sign waivers to turn over code – really don’t want this. But everyone has a different philosophy. Customers: If you pay for it with your $, you own it. But strongly emphasize that you contribute it to OS.

MG: Agreed. BSD, MIT, Apache are all good. GPL is problematic.

WS: Agreed. We go for permissive because it makes companies more comfortable. GPL can make companies very nervous.

SA: Used to be common to get a contract that said, “you are not allowed to use OS”. GPL gave OS a bad reputation. Lawyers just put blanket statements in the contracts. Needed to educate them on the difference.

CD: How you work with academic collaborators? Is geographic distance a challenge?

WS: Have an R01 with U Conn. We focus on engineering and academics work on the “big ideas”. Works well because we respect eachother’s skills. Because software development is so distributed these days, there are lots of tools to support this. Pretty seamless. But try to do a hackathon or project week regularly for some F2F time.

SA: Impossible to get an NIH R01 on your own these days. Most of our funding now is at the R01 level, both PI and subcontractor.

CD: What kinds of grants are you on? Traditional hypothesis testing? Or just technology development?

SA: Hypothesis-driven research. Eg scoliosis detection in children with non-ionizing radiation. Developing the algorithms and technologies for this. Use of OS tools greatly increased productivity of his academic lab.