Growing an Open-Source Community:
Lessons Learned from

Patrick Cozzi
@pjcozzi
pcozzi@agi.com
What is Cesium?

An open-source JavaScript library for world-class 3D globes and maps

http://cesiumjs.org/

Images from http://cesiumjs.org/demos.html
Traditional advice is to go open as early as possible to build a grassroots community. In practice, we didn’t build community until after Cesium was open-source for awhile.

Cesium has been open-source for ~4 years. 1.0 for ~2 years. We just announced a product this week.

Growing the user and contributor community took years, but it is like a snowball going down a hill, and gets easier.

See http://cesiumjs.org/2014/07/15/Milestones-Leading-to-Cesium-1/
Stats as of April 20, 2016.

Successful growth, but not unattainable.

Given the large A&D user base, GitHub stars, for example, do not reflect nearly the full size of the community.

The community has millions of users. NORAD Tracks Santa alone reaches 20 million users. Each month Flightradar24.com is visited more than 45 million times (not all using 3D!).

In addition to the forum, Stack Overflow and GIS Stack Exchange are used a bit.
Strong GitHub growth for Cesium 1.0. Forum traffic followed.

High variance in monthly posts. One active person can create a big difference. AGIers are also still the top posters. Dips during the holidays and graduation season.

GE plugin deprecation announcement - http://googlegeodevelopers.blogspot.com/2014/12/announcing-deprecation-of-google-earth.html
See https://github.com/AnalyticalGraphicsInc/cesium/graphs/contributors

There are a few more contributors who commits are still on a branch (i.e., 3D Tiles) that was not merged to master yet.
November 2012 - Cesium was released as open-source in April 2012. It didn’t get its first community contribution until November! 7 months!

March 2013 – GSoC (5 contributors) – small one-time contributions. Looks great on paper. Did not add much value to the project other than activity.

March 2015 - GSoC (2 contributors)

August 2014 – Community chips in to release Cesium 1.0

December 2015 – GE deprecation. Start of significant increased community contribution.

October 2014 – AGI’s Cesium Lab forms. Six months then team ramp-up. +2 this May.
Contributor Stats

- Contributors per month
CIS 399 – 2 Penn students

Initial Cesium Lab was five developers.

Realistically, we get a few community contributors a month, working on their own timeline, which is hard to predict.
Other AGI teams – like “internal open source”

https://github.com/AnalyticalGraphicsInc/cesium/graphs/contributors?from=2016-03-20&to=2016-04-20&type=c

https://github.com/AnalyticalGraphicsInc/cesium/graphs/contributors?from=2016-02-20&to=2016-03-20&type=c
New CLAs per year – not necessarily new contributors; not everyone who submits a CLA ends up contributing.

2013 and 2015 – each had six individual CLAs from GSoC. In 2013, 5 of the 6 contributed; in 2015, 2 of 6 contributed.

Corporate CLAs may look low compared to Individual CLAs. Often developers from organizations/corporations will sign individual CLAs – OK with us as long as they have permission from their organization.

Organization may also sign a CLA in case they contribute. The contribution may come six months later. A large organization may also take six months to sign the CLA.

Individuals generally contribute and sign at the same time. The trend is that more individuals who sign a CLA also contribute, 100% so far in 2016.
How did we get here?

How can your project get here?
With GitHub, it is so easy to contribute, that committer vs. contributor matters less than it use to.

A better measure of a project’s activity than total number of contributors is the number of contributors in the past 30 days. For example, on 11/05/2015, Cesium has 62 contributors, but only 25 in the past 30 days (still quite a high ratio).

In the very early days of Cesium, I focused too much on “grow the contributor community”, when I should have focused on “grow the user community”, because the contributor community comes largely from the user community who need bug fixes, believe in the project, etc.
What is a Contribution?

- Code, of course
  - Tests
  - Example code
  - Ecosystem projects
- Documentation and tutorials
- Sample data
- Forum participation
- Success stories and evangelism

Sample data, e.g., 3D models for testing glTF.

It is awesome – and more credible – when someone not involved in Cesium gives a talk at a conference on the benefits of Cesium.
The #1 reason developers outside of the core team contribute to Cesium is because they need a feature or bug fix.

Examples:
- New formats: TMS, URL template provider.
- Optimize: model cache, triangulation.
- Good open-source citizen: Cesium 1.0, doc fixes.
- GSoC, etc.: KML, GPX, CZML

The pain/cost/time of maintaining a fork is reason enough to contribute to a project.

Developers probably also want the experience and to learn, but I haven’t seen this much in my experience.

More details and specific examples in the bonus slides at the end of this talk.
Doc can take months, and then is hard to keep up to date. It is worth it because it is a high leverage activity.

Also good for new hires ramping up.

Trend – accept less than perfect, fix it up at first, and then ask for a contributor to more carefully follow the standards once they are more engaged. This keeps the barrier-to-entry low. Node.js promotes this. We are so busy with Cesium that this is hard for us to do.

Competition often says away from your core competency. They often want to collaborate instead of compete.
Provide Entry Points for New Contributors

- Make entry points easy to find

- glTF #456
Design for Contribution

• “Plugin” points
  – Cesium imagery providers, data sources, geometries, ...
• Clean and simple design and code
• Documentation
Response at least every day. Every hour if you can. Most people are too busy.
We tweet everyone’s first contribution.
Feature Success Stories

http://cesiumjs.org/for-google-earth-developers.html
Also wrote two tutorials, answered 100s of forum questions, and did related features and bug fixes.
Contributions Beyond Code

• Former Google Earth users advocate for Cesium
Gives a chance to engage the community: forum, twitter, blog. Also tweet upcoming features throughout the month.
Many users already know what licenses they can use.

Big corporate users in particular will carefully evaluate third-party libraries.
In pseudo-order of usefulness.

New Cesium releases – we pin these tweets; they among our most popular.

Upcoming Cesium features – “the making of” screenshots, animated gifs, links to pull requests, etc.
One forum at most – funnel the traffic to one place. “Appear as many”

Answering a forum question is a contribution.

Better for the community to answer a question on the forum, not in private email. It creates an achieve that has long-term value for many.

Corporate involvement – widely accepted today. Contributors should still retain copyright (CLA).

Some contributors outside of AGI also have commit access to Cesium.
Slow forum day can be 20 minutes. Busy day can be two hours. Timebox it.
Given that the project now has to maintain your code, they may not want some code changes, e.g., a 10% performance gain at the cost of significant complexity.
Reading

- [Producing Open Source Software](#), Karl Fogel
- [Art of Community](#), Jono Bacon
- [Hints for successfully managing an open-source project](#), David Catuhe, BabylonJS
- [Healthy Open Source](#), Mikeal Rogers, Node.js
Sign in and vote at foss4gna.org

Evaluate the Sessions

-1  0  +1
Thank You!

Contact me
Patrick Cozzi
@pjcozzi
pcozzi@agi.com
Bonus Slides
Big Ideas

• Users become contributors... because it helps them
• Have enough documentation... contributors will follow it
• Provide entry points... that are easy
• Design for contribution
• Respond promptly... to get contributors engaged
• Make contributors rockstars
Why is Cesium Open Source?

• Came from dev team, not business folks
• Pitch influenced by Innovation Happens Elsewhere
• Pitch
  – 3D is commodity (we were wrong!)
    • Give away commodity to sell value-add
  – Broaden the market and sell into new verticals
  – All successful JavaScript libraries are open source
  – It will be new, fun, and exciting
  – Get external contributions
  – Now able to use open-source channels for outreach
Why do Developers Contribute?

• It’s their job
• Beyond the core team...
Why do Developers Contribute?

• Need bugs fixed for their own project
  – Examples: doc fixes, geodesics
Why do Developers Contribute?

• Need features for their own project
  – Support their/new formats
    • Examples: TMS, URL template imagery
  – New version of a format
    • Example: terrain normals (internal user)
  – Missing features
    • Example: fromCrossProduct (internal user)
  – Support their build system
    • Example: browserify, webpack
Why do Developers Contribute?

• Optimize their use cases
  – Examples: model cache, triangulation
Why do Developers Contribute?

• Want to be good open-source citizens
  – Examples: Cesium 1.0, doc fixes
Why do Developers Contribute?

• Want to do business with us
  – But don’t want us to say more
Why do Developers Contribute?

• Required for GSoC, SOCIS, CIS 399
  – Examples: KML, GPX, CZML
Why do Developers Contribute?

• Want to apply for a job
  – GSoC. Surprisingly, have not seen this for full-time positions
Why do Developers Contribute?

• Don't want to maintain a fork
Why do Developers Contribute?

• Doc, tutorial, showcase, guest blog post
  – Want to build a relationship with us
  – Want the press
Bonus: Tips to get Contributions

• Grow the user community to grow the contributor community
• Make contributors rockstars - blogs, forum, twitter, etc.
• Use CLAs
  – High barrier to entry at first, now we get 1+ a week.
• Document standards, most contributors read them
• “Plugins” get most of the contributions
• It's really hard to find time to review contributor pull requests that aren't a priority, especially big ones
• Only merge code you are willing to maintain it
Beyond CLAs, also see A Legal Issues Primer for Open Source and Free Software Projects - http://www.softwarefreedom.org/resources/2008/foss-primer.html
Contributor License Agreements (CLA)

- Terms under which IP is contributed to a project.
- Used by Google (Chrome, Android, ...), Facebook, Open-Source Foundations, and all major projects I know of
- Protects you, the project, and the users
- Example terms:
  - “The project can use this code; I’m allowed to contribute it; I keep the copyright; I provide a patent grant”
- Apache Foundation CLA
  - Corporate: https://www.apache.org/licenses/cla-corporate.txt
  - Individual: https://www.apache.org/licenses/cla.txt

Google - https://cla.developers.google.com/clas
Facebook - https://code.facebook.com/cla

Every project should have a CLA
Today alone (11/06/2015), we received two CLAs.

CLAs

• Are CLAs a barrier to entry?
  – In Cesium, CLAs were hard to get signed at first
    • Big companies took months
  – Now that Cesium is established, CLAs come in all the time
Developer Certificate Of Origin (DCO)

• Signed-off commits
• Used by the Linux kernel
• We tried it in Cesium. No one used it


Also check out CLA assistant - [https://cla-assistant.io/](https://cla-assistant.io/)

And this interesting alternative to CLAs - The Berneout Pledge - [http://writing.kemitchell.com/2016/04/18/Berneout-Pledge.html](http://writing.kemitchell.com/2016/04/18/Berneout-Pledge.html)
Pull Requests without CLAs or DCOs

- Is a pull request good enough?
  - Yes, for something that isn’t IP, e.g., a typo fix
    - See https://cartodb.com/contributing/
  - No for IP since it doesn’t ensure the contributor had the right to contribute

The code could have came from elsewhere.

It might be the contributor’s employers code.
CLA Tips

• Ask your employer before you sign
• Use them in your project
  – CONTRIBUTING.md
  – Kindly ask contributors to sign

https://github.com/AnalyticalGraphicsInc/cesium/blob/master/CONTRIBUTING.md
Why Contribute?

(for students)
Fill a Real Need in a Visible Way

- Advice from Rich Geldreich (Unity, formerly Valve and Microsoft)
  - Find (or create) an open source project that game devs use, and contribute to it in a very visible way. For example, we [Valve] recently hired the author/maintainer of the SDL library. Google actively recruits devs who write libraries they use. Intel hires open source driver devs that contribute to MESA.
  - Identify a need and fill it with an open source project (that uses a license liberal enough that devs can actually use it). For example, right now a lot of teams are trying to target OpenGL ES, OpenGL, and D3D9/11 and are struggling with how to write a single set of shaders that can be translated/compiled to all these targets.
34 responses from CIS 565 alumni over six semesters (5 of which Patrick taught). The course is taught about once a year and has ~20 students. For more details, see: https://github.com/pjcozzi/Articles/raw/master/SIGGRAPH/2015/Preparing-Students-for-Industry-Using-Open-Source-and-GitHub.pdf

Keep in mind, it didn’t hurt the students who said No, and it helped 51.5%. I expect this to only increase as recruiting practices continue to change.

Several students also already had jobs before starting the course or received full-time positions based on their internships.
This quote is representative of many responses:

- The project portfolio is useful: demos, videos
- Employers generally don’t look at the code (or the student doesn’t realize it)
- The interview itself is what matters
Being asked about your own code is a great way to stack the odds in your favor.

I expect more companies to ask for code samples; as a hiring manager, I do.