

**P** Pavel Vodenski is nominated for promotion in this cycle.

## Achievement/Project

1

### Design and implement Android SDK & Platform

Architect

2016-07-01 /2017-02-28

**Impact:** Critical

#### Project Description:

Proxy's overall mission is to make robotics accessible to non-roboticists by providing an integrated hardware and software platform based on Android, with a high-level SDK and tools. I designed, implemented much of, and led the team responsible for the SDK. The SDK is a critical-impact component of Proxy's overall strategy: we measure our overall success via applications written on top of the SDK such as Snacky (a delivery bot) and Tidybot (a desk cleaner) and it's essential that the SDK work reliably and facilitate efficient development of these apps from the prototyping stage through to maturation.

See: Slides from our Q3-'16 [deep dive](#) and the [design doc](#)

Using lessons learned working on Glass, I designed a modular, loosely-coupled, testable, maintainable architecture to foster significant contributions from the rest of the team regardless of previous Android experience. This approach to software design has had a significant multiplier effect on the productivity of the rest of my team. The framework I've created has allowed the rest of the team to make significant contributions using best practices and reduced the Android learning curve, especially w.r.t. to the design of IPC services.

#### Summary of key contributions:

- Created an approachable software architecture which allowed a dozen team members with a variety of Android experience levels (including no experience at all) to all make meaningful contributions to the SDK and implementation: [go/proxy-android-contributors](#)
- At least [61 distinct apps](#) written so far in google3 (including experimental)
- Solved difficult problems such as
  - [transferring large data over Android's IPC mechanism](#)
  - fixing USB audio in/out on our Android branch from NVidia

Reviewer/Role

Feedback

**Max Braun**

(MANAGER)

7 / Senior Staff

Software Engineer

**Familiarity:** Very familiar

**Acorn Pooley**

(PEER)

6 / Staff Software

Engineer

**Role:** collaborator **Familiarity:** Somewhat familiar

The SDK a very important part of our project, and challenging to make easy to use while still expressive enough to solve challenging problems in robotics. Pavel has done a great job at addressing these conflicting requirements and coming up with a system for writing robot software. He is very responsive to issues and feature requests and is able to quickly address new requirements that arise.

**Benjie Holson**

(PEER)

4 / Software Engineer III

**Role:** "Apps Sub-Initiative" Lead **Familiarity:** Very familiar

Pavel is the owner, architect and primary author of our primary SDK which is used daily by a dozen people. Our success as a moonshot is measured in the success of applications build on top of the SDK (Snacky & Tidier). His work has been critical in making a stable, simple SDK layer to create these apps. I had no Android experience prior to starting this project and Pavel's design allowed me to jump in and immediately be productive. In addition he is always available and happy to mentor, advise and help me with questions related to Android or the SDK.

<b>Elmar Mair</b> (PEER) 5 / Software Engineer	<b>Familiarity:</b> Not at all familiar
<b>Gabriel Cohen</b> (PEER) Director, Product Management	<b>Role:</b> product management lead <b>Familiarity:</b> Very familiar Proxy is at its core a hardware and software development platform for building robotics applications, so our SDK is the surface of the product to our users (developers). Pavel designed and implemented this critical top layer of our stack and oversees a broad group of contributors to it. He was able to translate the idea of an autonomous programmable robot into this fully functioning and now mature system supporting a large group of developers.
<b>Hans Peter Brondmo</b> (PEER) Director, Product Management	<b>Role:</b> GM Proxy <b>Familiarity:</b> Very familiar The L5 SDK is indeed a critical part of the Proxy SW stack and central part of our overall architecture. Pavel's technical and team leadership has been pivotal to the success of the project so far.
<b>Matthieu Guilbert</b> (PEER) 6 / Staff Roboticist	<b>Role:</b> User of the SDK and Platform <b>Familiarity:</b> Very familiar I am a non android app programmer and I used the proxy Android SDK and Platform and this was a fantastic and smooth experience, it took me less than 2 hours to: install the SDK, get the first "Hello World" program and have my first robot roaming around running what is now called continuous operation. One of the major problem that this layer is facing is simplicity and Pavel nailed it. He is the sole designer of this piece of work, it was extremely ambiguous what needed to happen, he was able to define the right design together with the right abstraction to have a 10+ team working on it. This is a part of the robotic stack that we can all be proud of ! Thanks Pavel.
<b>Paige Hinkle</b> (PEER) 3 / Software Engineer II	<b>Role:</b> Engineer <b>Familiarity:</b> Very familiar Pavel's leadership and technical contributions simplified the complexity of writing robot applications, which is essential to our success as a team. He is a frequent advocate for the Android ecosystem and makes sure that we are providing our developers with a good experience. Historically our robot apps have been difficult to implement, test, and maintain. Pavel's SDK work makes it possible to write organized, intuitive robot apps.
<b>Rainer Hessmer</b> (PEER) 5 / Senior Software Engineer	<b>Familiarity:</b> Somewhat familiar I can only comment based on my experience with the SDK over the last three month. I am impressed by the very clean separation of concerns into libraries and services. It makes climbing into the code base fairly easy. I experienced the impact myself but maybe more importantly, I saw how quickly the intern <a href="#">hannalee@</a> was able to leverage and extend the existing infrastructure to add support for monitoring CPU / GPU loads reported via IPC (see e.g., <a href="https://critique.corp.google.com/#review/162510852">https://critique.corp.google.com/#review/162510852</a> ).
2 <b>Stabilize and simplify Android SDK &amp; Platform</b>  Architect 2017-03-01 /2017-08-18 <b>Impact:</b> Critical	<b>Project Description:</b> During Q2 and Q3 of this year, I focused my Android Platform (the implementation of our Android SDK and tools) work on improving stability. In Q2, feedback from key stakeholders showed that platform instability was a major pain point. I took a multi-pronged approach to measure and improve stability. Qualitative and quantitative signals indicate that this was a success—stakeholders perceive and data show that Android platform stability is improved as a result of my work.  Last cycle, I identified my lack of hands-on experience with our own SDK as an opportunity for more impact. In this cycle, I addressed that and identified opportunities to simplify and solve problems for our application developers and thereby improve overall Proxy velocity. I simplified a core (and, therefore, pervasive) pattern in our SDK, removing a common opportunity for user error and confusion and barrier to unit testing. I also designed and adopted a pattern to solve a common challenge in writing modular, interactive applications. As such, I proactively identified SDK pain-points and emerging usage patterns and reduced chaos from the system using my deep expertise in refactoring and software design.  I also evangelized the Experience initiative to the rest of Proxy to help with overall visibility and cohesion. I also exhibited qualities of an exceptional IC (even given Google standards) by excelling "in the clutch"—although diligent, grungy work in the preceding quarters made this saving throw possible.

## Summary of key contributions:

### Platform stabilization

- Identified, characterized, resolved platform instability
  - [Tracking doc](#), shared with key stakeholders
  - [Redesigned lifecycle of platform process](#) to allow graceful recovery from uncaught exceptions
  - [LauncherService](#), a critical and initially difficult-to-maintain component, got a lot of my attention
    - [Added event monitoring](#) around the key use case of starting applications, worked to drive and keep down failures
      - App launch success rate has been [95% average for the past month](#) (91% worst week), up from 83% (73% worst week) during June
    - Fixed critical [user-visible bugs](#)
    - [Refactored the service](#) for simplicity and testability
    - Integrated LauncherService [error handling with the Companion app](#) and (with [dalam@](#)) with text-to-speech feedback to make errors actionable
- [Improved accuracy of and simplified robot readiness-for-use monitor](#)
- Thanks to [nacorn@](#)'s implementation of streaming support, leveraged [GRPC streaming for camera and perception data](#) to reduce CPU usage and improve stability of both the Android layer and the underlying robot API
- The culmination of this effort has been that, of [23 errors logged over 6 hours of monitored Tidybot runs in Q3](#), platform instability only accounted for precisely [one](#)
- "Fix L5 stability" is a large, open-ended problem without a singular clear solution; I implemented a multi-pronged approach using a combination of monitoring, testing, refactoring, and user-facing changes. Doing so has had a noticeable positive impact on Proxy velocity, especially w.r.t. to Tidybot, our primary focus for H2 of 2017.

### SDK simplification

- Vastly simplified the SDK by [migrating all of our API calls from a monadic pattern wherein the API call ret...](#)
  - Proactively identified that
    - the blocking/asynchronous flexibility intended by the initial monadic pattern was not meaningfully exercised by our developers
    - users (from novice to expert) routinely forgot to execute the resulting block -- a frustrating, silent error unpreventable by the compiler without annotations and presubmit checks
    - the pattern created a significant barrier to unit testing
  - Used decision-making by consent to ensure buy-in from clients before starting
  - Used my expertise in behavior-preserving refactoring to implement this migration incrementally over 24 CLs
    - This ensured stability (no migration-related regressions) and kept CLs manageable for reviewers
  - Simplified unit testing allowed us to quickly get [50% line coverage for Tidybot](#)
- Designed and implemented a BehaviorManager library
  - Identified that our SDK is missing a pattern for modular, user-interruptible application development
    - Our developers typically write imperative applications at the prototype stage and later layer in dynamic user-interruptibility (e. g. using a hotword speech recognition, or human perception signals from body language)
  - Experimented with this sort of application during our App Development Mondays sessions, including by evaluating [Kotlin co-routines as an agent-centric approach](#)
  - Took inspiration from Android Fragments for composability and modularity
  - Implemented BehaviorManager as [a well-tested optional add-on to our SDK](#), wrote [example code](#), socialized the idea with our developers, and migrated our core user agent to use it
  - BehaviorManager is our top-candidate approach for layering human interaction into Tidybot, the Proxy top priority for H2 '17

Reviewer/Role

Feedback

<b>Max Braun</b> (MANAGER) 7 / Senior Staff Software Engineer	<b>Familiarity:</b> Very familiar
<b>Acorn Pooley</b> (PEER) 6 / Staff Software Engineer	<b>Familiarity:</b> Not at all familiar
<b>Benjie Holson</b> (PEER) 4 / Software Engineer III	<p><b>Role:</b> Apps Sub-Initiative Lead      <b>Familiarity:</b> Very familiar</p> <p>Pavel's Platform stability effort has been very successful and very impactful. Previous we were in a quagmire of breakages across the stack with little visibility as to what happened. Working at the SDK layer the Android/launcher layer often took the blame for instability whether or not it was the root cause. By adding good tracking Pavel was able to move the needle not only the stability of the Android platform and Launcher Service but across the stack. Writing apps feels qualitatively different than it did before this effort: before if the robot didn't work I would first restart the android platform and then restart the set of robot control services and try again. Now I expect those to just work and look first to my own code. Also, before this effort restarting the Android platform service was something that I did multiple times a day (not necessarily because it was needed, but because it was easy and sometimes helped). I have only needed to restart it once in the last month.</p> <p>Pavel's stability success and SDK design was a big part of why we were able to port Tidy from C++/Linux to Java/Android in a single week and run essentially error free a day early.</p>
<b>Elmar Mair</b> (PEER) 5 / Software Engineer	<b>Familiarity:</b> Not at all familiar
<b>Gabriel Cohen</b> (PEER) Director, Product Management	<p><b>Role:</b> product management lead      <b>Familiarity:</b> Very familiar</p> <p>Pavel was very responsive and then proactive in chasing platform stability issues to ground. I remember him telling me in a 1-1 that he decided to enter a personal code yellow to get stability into a good place. I was really impressed with how pragmatic and how organized he was in pursuing this. He stood up additional logging and metrics, consulting with all the key stakeholders, and just burned it down until it simply wasn't an issue any more. It's wonderful to Pavel model the behavior of taking on grungy but necessary work and pursuing it with gusto.</p>
<b>Hans Peter Brondmo</b> (PEER) Director, Product Management	<p><b>Role:</b> GM Proxy      <b>Familiarity:</b> Very familiar</p> <p>Pavel's contributions to and leadership of the design and implementation of the SDK has been and continues to be critical to Proxy's success.</p> <p>His work on stability and reliability has greatly increased the productivity of s significant portion of the SW team.</p>
<b>Matthieu Guilbert</b> (PEER) 6 / Staff Roboticist	<p><b>Role:</b> Tech lead of SW Foundation      <b>Familiarity:</b> Very familiar</p> <p>As all new piece of software, SDK and platform needed a stabilization time. This year really showed the super powers of Pavel, he was able to:</p> <ul style="list-style-type: none"> <li>- stick to the current design because it was good</li> <li>- put in place the right tools to analyze the problem</li> <li>- do the necessary grungy work to write the tests and fix the problems</li> </ul> <p>Another super power of Pavel during this work was the ability to build consensus around him. And at the end of the day the quantity of work that Pavel was able to put together is actually completely amazing ! Look at his number of CLs and lines of code that he wrote !</p> <p>By the way, all this code is running all day long during our continuous operation runs and used by the full Proxy team</p>

<b>Paige Hinkle</b> (PEER) 3 / Software Engineer II	<b>Role:</b> Engineer <b>Familiarity:</b> Very familiar Before Pavel's stabilization work, our robots often crashed in mysterious or unpredictable ways. Now engineers on the Proxy team are able to work much faster and focus on their work instead of dealing with the frustration of an unstable robot. Launcher Service was incredibly difficult to get right because of the variety of ways that robots can fail, but because of Pavel's work we now have an much more consistent launch process. Behavior Manager is a huge improvement for writing modular robot apps. This was a difficult software design challenge that Pavel recognized and provided a solution for.
<b>Rainer Hessmer</b> (PEER) 5 / Senior Software Engineer	<b>Familiarity:</b> Not very familiar
<div>3</div> <b>Companion App &amp; SDK Tools</b>  Architect, Advisor 2016-08-15 / <b>Impact:</b> Critical	<p><b>Project Description:</b>          As an architect, prototyper and implementor, I've led Proxy to create a maintainable suite of developer tools to accompany our robot and SDK. Unlike a typical robotics suite (such as ROS), this suite is built on Android and the web, which:</p> <ul style="list-style-type: none"> <li>• makes our tools portable and facilitates iterative development; there's nothing to install (except the Companion app, which auto-updates) and no need to worry about compatibility with the host OS nor cross-compatibility with software on the robot</li> <li>• lets us leverage google3 and existing Google processes (c. f. the challenge of building desktop applications on top of Google infrastructure)</li> <li>• makes it easier to bring on senior developers as internal transfers rather than needing to find external hire for native desktop development</li> </ul> <hr/> <p><b>Summary of key contributions:</b>  <b>Companion App (Q3 '16, handed off to <a href="#">phinkle@</a>)</b>          I wrote the <a href="#">initial version of our companion app</a> as well as a <a href="#">prototype cloud backend</a> (and see the <a href="#">one pager</a>). This is invaluable to our team for several reasons:</p> <ul style="list-style-type: none"> <li>• Grounded our discussion about a user interaction model for robot apps, including in a many-users-to-many-robots environment</li> <li>• Vetted Firebase as a cloud prototyping framework for the team; clarity around the "cloud story" for L5 is in demand by leads across initiatives</li> <li>• Set up our new team member <a href="#">phinkle@</a> with an opportunity to quickly get up to speed with Android in the short-term and to own and drive a substantial project in the longer-term</li> </ul> <p><b>Web Visualizer (Q4 '16/Q1 '17, with <a href="#">easchwar@</a>)</b></p> <ul style="list-style-type: none"> <li>• Prototyped WebRTC data channel as the communication layer for our in-browser visualizer</li> <li>• Designed a WebRTC signaling mechanism on top of Firebase and the internal Network Traversal API (TURN cloud)</li> <li>• Implemented the robot side of the WebRTC <a href="#">communication pipeline</a></li> </ul> <p><b>Annotator (Q1 '17, with <a href="#">phinkle@</a> and <a href="#">easchwar@</a>)</b></p> <ul style="list-style-type: none"> <li>• <a href="#">Designed</a> a cloud-based object teaching tool, defining the system architecture across the browser tool, the application layer of the robot, and the underlying robot APIs</li> <li>• Implemented the <a href="#">annotator data model</a></li> </ul> <p><b>Flock, Map Manager (advisor/backend only)</b></p> <ul style="list-style-type: none"> <li>• Provided underlying data for <a href="#">prisament@</a>'s implementation of <a href="#">Flock</a></li> <li>• Provided backend and guidance for Ischeurmann's implementation of <a href="#">Map Manager</a></li> </ul>
<b>Max Braun</b> (MANAGER) 7 / Senior Staff Software Engineer	<b>Familiarity:</b> Very familiar
Reviewer/Role	Feedback

<b>Acorn Pooley</b> (PEER) 6 / Staff Software Engineer	<b>Familiarity:</b> Not very familiar
<b>Benjie Holson</b> (PEER) 4 / Software Engineer III	<b>Role:</b> Daily user of the tools <b>Familiarity:</b> Somewhat familiar  Web Visualizer is a a critical tool that makes debugging issues on robots possible. The bandwidth requirements, distributed nature of the producers of the data (it is produces by any of the three computers on the robot) and the variability of the kinds of data makes the transport for the visualizer a knarly technical problem.  The Companion app made it possible to run snack delivery across all of RLS without lugging a laptop around. We were able to start, debug, cancel and resume directly from our phones.
<b>Elmar Mair</b> (PEER) 5 / Software Engineer	<b>Role:</b> Team lead and user <b>Familiarity:</b> Somewhat familiar  In my team, which works on robot localization, mapping and base motion, we heavily depend on the Map Manager, Flock and the Web Visualizer. All three tools are crucial for our development. The decision to use web frontends allows us to use these tools on any platform (desktops, laptops, phones). Pavel is a relentless driving force behind all these tools. Furthermore, he has a great in depth understanding which allows him to help out if crucial fixes or feature requests are required. He is very responsive and keeps the big picture in mind when discussing changes. He aims for general but simple and intuitive interfaces.
<b>Gabriel Cohen</b> (PEER) Director, Product Management	<b>Role:</b> product management lead <b>Familiarity:</b> Very familiar
<b>Hans Peter Brondmo</b> (PEER) Director, Product Management	<b>Role:</b> GM Proxy <b>Familiarity:</b> Very familiar  More than anything, Pavel's strong initiative, design and architecture focus, hands on technical leadership and strong ability to work with jr and senior engineers alike in order to build consensus and gain leverage is illustrated by the several projects (outlined above) that he's either identified, started and handed off or made major contributions to.
<b>Matthieu Guilbert</b> (PEER) 6 / Staff Roboticist	<b>Role:</b> User of all those tools <b>Familiarity:</b> Very familiar  yet another fantastic quality of Pavel, he can manage a team of fairly junior SWE and get professional results: <ul style="list-style-type: none"> <li>- The companion app is used daily by most of the team (I use it daily and is simple, easy to use and very useful)</li> <li>- Web visualizer still needs a lot of work but the design allows fast streaming of images</li> <li>- The Map manager is a fantastic tool (still new) that will speed up all map development</li> </ul> Pavel really was able to delegate work and this was easier because of the fantastic design of this layer together with a great leadership of Pavel.
<b>Paige Hinkle</b> (PEER) 3 / Software Engineer II	<b>Role:</b> Engineer <b>Familiarity:</b> Very familiar  Pavel's work with SDK tools has helped our team become much faster and more effective. His ability to jump-start projects and write quick prototypes has saved many engineering hours. Pavel's mentorship and knowledge of software design has been critical to all of the applications that he mentions above. He has helped me with difficult problems related to the Companion App that I could not have solved without his guidance.
<b>Rainer Hessmer</b> (PEER) 5 / Senior Software Engineer	<b>Familiarity:</b> Not very familiar
4 <b>Proxy Cloud Platform</b>	<b>Project Description:</b>

Architect  
2017-04-03 /2017-08-18  
**Impact:** Critical

In Q3 of '16, I stepped up to pioneer Proxy's cloud effort, filling a pressing need. My work to leverage and evangelize for Cloud Platform, Cloud Endpoints and Firebase has allowed junior engineers within my own team and engineers elsewhere in Proxy to prototype and build cloud applications with minimal effort. Despite my initial minimal experience in this arena (especially at Google), I've solved critical needs such as user authentication, robot identity, remote application launch and monitoring, peer-to-peer visualization, real-time analytics, and real-time sharing of map data.

#### Summary of key contributions:

##### Identity, Launcher, Visualization, Teaching

- Created infrastructure for distributed data across our robot Android application layer, Android companion application, web tools, and command line developer tool. All of these tools use Firebase via the ProxyFleetControl project and associated frameworks. Other engineers and interns have been able to leverage this platform to quickly prototype and launch robot productivity applications.
- Designed and implemented the [identity framework](#) to authenticate robots without GAIA accounts. Worked with Federated Identity and Firebase teams on the design.
- Owned authentication integration for the `fleet` CLI (<https://critique.corp.google.com/#review/144266452>); added ability to [easily sign robots in to PFC with fleet](#).

##### Analytics

- Designed and implemented [analytics pipeline](#) using Firebase Realtime Database, Firebase Cloud Functions and BigQuery. This system has subsumed 62M events in six months.
  - This serverless architecture is a great fit for Proxy, which is short on backend/cloud infrastructure engineers
- Wrote client APIs for Android SDK and Platform
- I used this system to implement data collection around key metrics for my team and others
  - [Application launch success rate](#)
  - [Hardware error rates](#) (for Proxy code yellow)
  - [Robot module failure rates and reasons](#) (for Software Foundation)
- Other engineers have used this system to implement logging around other key metrics/scenarios across Proxy:
  - Total distance driven ([hwestao@](#), [dmillard@](#))
  - CPU/Memory utilization ([hannalee@](#))
  - Build and deploy latency ([rainerh@](#))
  - User Interaction outcomes in Snacky and Tidybot ([holson@](#), [gbudd@](#))
- Handed off project to [rainerh@](#), continuing to collaborate/advise

##### Map Sync

- With [elmarm@](#), designed and implemented a system to share map data in real time across our fleet
  - Allows developers to implement applications in terms of abstract map data instead of hard-coding coordinates w.r.t. a static map.
  - Robot fleet can autonomously update its map, maintaining the appropriate geospatial relationships between reference locations.
  - Facilitates data-driven applications; instead of recompiling applications solely to e. g. tweak the position of a location of interest, developers can use a web tool or even robot SDK applications to add/update map data and change behavior dynamically.
- I implemented the cloud backend and the on-robot system that connects to the cloud backend, updating the underlying robot mapping modules with new map data in realtime
  - [Cloud endpoint implementation](#)
  - [Sync adapter implementation](#)
  - [SDK implementation](#) and [example code](#)
- With my support, [lscheurmann@](#) was able to use the map cloud API to write a [Map Manager](#) tool to allow developers to work with map data
- This system has been [integrated into Tidybot](#) (by [hwestao@](#))

Reviewer/Role

Feedback

<b>Max Braun</b> (MANAGER) 7 / Senior Staff Software Engineer	<b>Familiarity:</b> Very familiar
<b>Acorn Pooley</b> (PEER) 6 / Staff Software Engineer	<b>Familiarity:</b> Not very familiar
<b>Benjie Holson</b> (PEER) 4 / Software Engineer III	<b>Role:</b> User of event logging. <b>Familiarity:</b> Somewhat familiar My interaction with this subsystem has been as a customer of the event logging system. As a client I barely had to think about it at all (which is exactly how you want it). Logging an event is a simple one liner and the events show up queryable and ready for dashboards.
<b>Elmar Mair</b> (PEER) 5 / Software Engineer	<b>Role:</b> Tech lead Mapping, user <b>Familiarity:</b> Very familiar I'm leading the Mapping efforts in the project. Having up-to-date maps on all robots is a crucial requirement for our goal of lifelong mapping which requires constant map updates. Pavel played a crucial role in architecting and implementing the cloud storage solution and the communication between cloud and robots. This is a very complex task considering that the solution requires authentication, synchronization, and has to scale for the full fleet. The implemented approach leverages features in Android, Firebase and Google Cloud Storage and it seems to become the basis for further, similar communication channels. A crucial tool for the team is the MapManager which is a web frontend which allows to visualize and modify map content. <a href="#">lschuermann@</a> developed the tool under Pavel's and my supervision. This tool would not have been possible without Pavel's help and it clearly carries his signature. Aside from the people mentioned, also <a href="#">cmansley@</a> and <a href="#">rphilipp@</a> are using Pavel's analytics framework to report metrics. <a href="#">cmansley@</a> even extend the framework to be used in lower layers in our stack. Having metrics is crucial and Pavel's framework became an essential part of our data driven development.
<b>Gabriel Cohen</b> (PEER) Director, Product Management	<b>Role:</b> product management lead <b>Familiarity:</b> Very familiar
<b>Hans Peter Brondmo</b> (PEER) Director, Product Management	<b>Role:</b> GM Proxy <b>Familiarity:</b> Very familiar
<b>Matthieu Guilbert</b> (PEER) 6 / Staff Roboticist	<b>Role:</b> User and TL of SW Foundation <b>Familiarity:</b> Very familiar This has been a key contribution of Pavel. Proxy and before kept talking about Cloud, etc. Pavel actually made it a reality, he took the first stab at it by creating a firebase database from which we could start collecting analytics and also having a common source of truth for robot state between our different Productivity tool such as Fleet / Flock and the companion app. The impact of this database is gigantic within Proxy, execution is not the strength of Proxy and this infrastructure allowed to gather data, create metrics and have dashboard. Proxy went from a fuzzy organization to a data driven organization and thus because of Pavel ! if Proxy executes better, it is greatly because of the great work of Pavel.
<b>Paige Hinkle</b> (PEER) 3 / Software Engineer II	<b>Role:</b> Engineer <b>Familiarity:</b> Very familiar Pavel was the initial architect of our cloud systems. His work enabled many other projects (Web Visualizer, Annotator, Flock, Map Manager, Fleet CLI improvements, Analytics) to communicate with the robot in an effective and easy way. His work on authentication was especially difficult and required coordination with multiple teams inside and outside of Proxy to get right.



**Rainer Hessmer**  
(PEER)  
5 / Senior Software  
Engineer

**Role:** Proxy Cloud TL (joined three months ago)      **Familiarity:** Somewhat familiar

I am very impressed by the breadth of the implementation and the careful selection of underlying technologies that Pavel accomplished essentially from scratch. E.g., tying together custom authentication and authorization with Firebase Realtime Database, Firebase Storage and AppEngine requires deep insight. The implementation of the Analytics pipeline from Android SDK to Firebase, Firebase rules, cloud functions, BigQuery proved to be highly scalable and provides very low latency analytics reporting. I was very happy to see that the decisions to use the chosen technologies were based on upfront investigation of other available technologies. E.g., for Proxy's analytics pipeline, first [Firebase Analytics](#) and Android's internal [Clearcut](#) tool were considered and tested. Both were discarded due to limitations that would hinder the Proxy use cases.

I took over the work on Analytics from Pavel. The handover was smooth and easy. Key contributing factors were:

- Clean code base with associated unit tests (e.g. the Android Platform's [analytics code](#) makes good use of dependency injection to allow tests against mocked backends ([AnalyticsServiceImplTest.java](#))
- g3doc documentation alongside the code base (e.g., [here](#) and [here](#))
- Pavel was readily available to provide verbal overviews including explanations and reasoning behind the chosen implementations.

5

## Leadership & Management

Manager, TL, Mentor  
**Impact:** Critical

### Summary of key contributions:

#### Roles & Initiatives

- TL (and since Q2 '17, TL Manager) of SDK/Tools team
- Android development and code health lead for Tidybot (Proxy-wide H2 top priority)
  - Implemented and evangelized iterative project management with [Kanban](#). Initially picked up technique from [craiglatimer@](#), used with my own team, evangelized to our Tidybot effort; trained other TLs and several other teams have picked up Kanban as a result
- With [holson@](#), created an initiative ("[App development Fridays](#)") to fishfood our SDK, foster knowledge transfer and cohesion among the team and organically drive SDK requirements. [holson@](#) has run this since Jan. '17
- Drove an initiative to [raise line coverage](#) for our Android codebase from 40% to 60% in December

#### Go-To & Visibility

- Go-to resource for all things Java, Android, google3 and cloud; point of contact for anyone at Proxy who deals with our application and cloud frameworks (engineers, QA, PM, etc.)
- Known as "technical center of gravity" for Proxy application layer; I've been somewhat prolific with [650 CLs landed and 1344 reviewed since July 2016](#)
- Shared periodic newsletter with all of Proxy to highlight SDK contributions from the rest of the team for visibility: [April](#), [May/June](#)
  - QPR in July and a team-wide ML incubator course in August have led to fewer new features since then
- Migrated entire [Proxy Android codebase from Nano to Lite protobuf in a day](#) to address unexpected build breakage due to nano deprecation
  - This was an exceptional circumstance, but validated my and others' insistence on thorough unit test coverage; despite touching ~3k lines across 191 files, this change only caused a single low-impact [regression](#)

#### Mentor

- Manage team of three L3s
- Hired and [onboarded](#) L5 SWE transfer ([irundquist@](#))
  - Wrote Grow posting, 7 GVC-screens with [braun@](#), coordinated and evaluated feedback from 3 on-site sessions
- Completed New Manager Flagship training
- Advised/assisted:
  - Interns: Ischeurmann, hannalee, skirmani
  - Engineers: rainerh, hwestao, sibigtroth, elmarm, yzgong, zhensong, dmillard, bhomborg, rjulian, phinkle, sarahcoe, dalam
- Host to highly successful engineering resident: [samgho@](#)

Reviewer/Role	Feedback
<b>Max Braun</b> (MANAGER) 7 / Senior Staff Software Engineer	<b>Familiarity:</b> Very familiar
<b>Acorn Pooley</b> (PEER) 6 / Staff Software Engineer	<b>Role:</b> customer <b>Familiarity:</b> Somewhat familiar Pavel has a great ability to understand varying requirements and come up with solutions that address them. He is flexible and willing to make changes when it is discovered that (minor or major) redesign will improve the system. He is also very good at working with people and aiding understanding of his and related parts of the system.
<b>Benjie Holson</b> (PEER) 4 / Software Engineer III	<b>Role:</b> Coworker <b>Familiarity:</b> Very familiar When I have a question about Android or Java best practices, Pavel is my first choice to go to. His expertise, thoughtfulness and humble consideration of questions makes him an extremely effective mentor.  App Day has become an integral part of the App Team's mission.
<b>Elmar Mair</b> (PEER) 5 / Software Engineer	<b>Role:</b> Team/tech lead, client <b>Familiarity:</b> Somewhat familiar Pavel is a key player in Proxy and a go-to person for anything related to Android, Java, and the cloud. He is not just a great architect with an eye for simple, generic solutions, but he also a strong SWE with clean and well tested contributions. He is having significant impact in the project and I don't know anyone who does not like working with Pavel.
<b>Gabriel Cohen</b> (PEER) Director, Product Management	<b>Role:</b> product management lead <b>Familiarity:</b> Somewhat familiar
<b>Hans Peter Brondmo</b> (PEER) Director, Product Management	<b>Role:</b> GM Proxy <b>Familiarity:</b> Very familiar
<b>Matthieu Guilbert</b> (PEER) 6 / Staff Roboticist	<b>Role:</b> TL of SW Foundation <b>Familiarity:</b> Somewhat familiar Pavel showed very good leadership: <ul style="list-style-type: none"> <li>- good design (helped with the separation of the work)</li> <li>- fantastic people management. It is always difficult to manage junior people and Pavel excelled at it !</li> <li>- he is the go-to person for all Android related questions and all robot application related question</li> </ul>
<b>Paige Hinkle</b> (PEER) 3 / Software Engineer II	<b>Familiarity:</b> Very familiar Pavel has done a great job as the SDK team lead and manager. He is very knowledgeable, good at communicating, and eager to help, which makes him a go-to person anytime somebody needs help or has a design question. As a TL, he makes sure that everybody is onboard before making any major technical decisions. His commitment to code health and test coverage is important, as our team moves very fast and instability slows everybody down.  As a manager, Pavel provides the right balance of support and autonomy. He is invested in his reports' careers, is quick to find solutions to technical and non-technical problems, and makes sure that everybody on the team is working on important, interesting projects. Pavel is also great at recognizing other peoples' work and making them feel valued.
<b>Rainer Hessmer</b> (PEER) 5 / Senior Software Engineer	<b>Role:</b> Cloud TL <b>Familiarity:</b> Somewhat familiar I want to particularly point out that in spite of all the work Pavel is doing, he has always been readily available to answer my questions or to provide deep dives into relevant parts of the implementation. This has been instrumental in allowing me to come up to speed quickly.

**Glass - Voice Input**

Tech Lead

2013-10-01 /2015-02-13

**Impact:** Critical**Project Description:**

I was the TL of Glass' voice input team, responsible for our first- and third-party voice APIs and UI.

**Summary of key contributions:**

- designed and implemented APIs for developer-defined voice actions and multi-step hotword/open-ended speech flows
- internationalized hotword voice action and contact menus
- led the team by writing design documents and work plans ([go/glass-voice-testing](#), [go/glass-voice-i18n](#), [go/glass-global-voice-menu](#)) and by setting and evangelizing a clear vision for voice input
- in the last two quarters on the team, grew my scope to cover all application-level audio (music and phone calls)

Reviewer/Role

Feedback

**Max Braun**

(MANAGER)

7 / Senior Staff

Software Engineer

**Familiarity:** Very familiar**Benjie Holson**

(PEER)

4 / Software Engineer III

**Familiarity:** Not at all familiar**Elmar Mair**

(PEER)

5 / Software Engineer

**Familiarity:** Not at all familiar**Gabriel Cohen**

(PEER)

Director, Product  
Management**Familiarity:** Not at all familiar**Matthieu Guilbert**

(PEER)

6 / Staff Roboticist

**Familiarity:** Not at all familiar**Paige Hinkle**

(PEER)

3 / Software Engineer II

**Familiarity:** Not at all familiar**Rainer Hessmer**

(PEER)

5 / Senior Software  
Engineer**Familiarity:** Not at all familiar**How well has Pavel Vodenski performed since your last assessment?**

Reviewer/Role

Feedback

**Max Braun**  
(MANAGER)  
7 / Senior Staff  
Software Engineer

Pavel has greatly exceeded expectations at L5 for multiple cycles and is measured best against the expectations at L6. His impact spans "a large work group, a very deep technical problem, and a long time horizon", as evidenced, for example, by building a scalable client/server infrastructure for Proxy robots from scratch, out of which came multiple engineers' full-time projects. Leading Proxy's Android SDK is an example that requires deep technical expertise, especially since it is the integration point for a very complex robotic system and comes with a high expectation of reliability. Pavel deals with "large, open-ended problems" such as defining the Android SDK, from the APIs to each of the developer tools. He makes "exceptional technical contributions" in the quality and volume of his code, the documentation of his code and his designs, and in following engineering best practices. Pavel "models strong teamwork and collaboration skills and has a noticeable positive impact on the velocity and success of his teammates" and "provides technical mentorship to members of his team by modeling high standards and helping other members of the team grow". This can be seen in the work done by his direct reports [phinkle@](#) (Companion App, Annotator), [sarahcoe@](#) (HRI), [kkleiven@](#), and previously [easchwar@](#) (Visualizer), but also in the wider team by [rainerh@](#) (Cloud), [holson@](#) (Apps), and [elmarm@](#) (Autonomy). He has above-average X Teamwork scores and received an unsung hero nomination. Pavel "actively reduces chaos from the system", such as the refactor of the Command<R> pattern. He "increases availability and reliability and makes data-driven optimizations and adjustments", e.g. with his successful SDK stability effort. Pavel's "leadership is evident across multiple groups", mainly in the sense of "influencing designs/plans/etc beyond just those of a single project or team", with virtually every Proxy engineer interacting with one of Pavel's projects every day.

**Why should you be promoted?**

Reviewer/Role	Promote?	Feedback
---------------	----------	----------

**Pavel Vodenski**  
(SELF)  
5 / Senior Software  
Engineer

Yes

I've shown over the past year that I'm already operating at an L6 level. The large, open-ended problem I've been solving is how to use Android to make robotics accessible to non-roboticists. To solve this problem, I've made exceptional technical contributions, mostly from a blank slate:

- Designed and implemented an Android SDK modeled after Google Play Services
- Designed and implemented an Android Companion app
- Designed and implemented a mostly-serverless cloud platform with support for
  - GAIA-less authentication
  - real-time data sync across robots and tools
  - real-time analytics events
- By virtue of the above, facilitated and worked with other engineers to create tools for fleet management, visualization, object teaching, and editing map data

Through the thoughtfulness of my designs, I've created an approachable, unsurprising application layer which has allowed engineers new-to-X and new-to-Android to make a variety of meaningful contributions. I've also eliminated complexity for our internal users; for example, my work on cloud and the Companion app makes connecting to robots as easy as clicking a "Sign In" button--no pairing over bluetooth or WiFi, no updating configuration files to add new robots to the fleet (and, therefore, no recompiling or waiting for a new release of the app). Through my work to create our real-time analytics system and to popularize its use, I've driven a culture of data-driven debugging and continuous improvement.

I've had various leadership roles on the team throughout the past year.

- I've been the go-to for all things Android, applications and Cloud. Our overall software stack divides up into layers from L0 (hardware) to L6 (cloud, tools). Other L6 SWEs are responsible for layers 0-4, and I created and am responsible for L5 (the application layer) and L6.
- I've led or kicked off initiatives such as to improve test coverage or to drive ongoing hands-on use of our platform (App Dev Fridays, now Mondays)
- I've mentored engineers from interns to senior SWEs, helping them get up to speed with or improve their understanding of Android, Google Cloud Platform, unit testing, software design, etc. I'm now managing three of those L3 engineers.

In my earlier life as TL for Voice on Glass, I drove and implemented our vision for voice UIs and APIs, leading a team of 3-4 engineers for 18 months.

By virtue of this exceptional IC work, coupled with my contributions as a mentor and manager, I believe I've demonstrated my readiness for promotion to L6.