Ashar Farooq

Engineering Practice Requirement Essay (EPR 2)

Spring 2023

Interview Report

I. Note on Interview

The 58:41 long interview was conducted on Sunday, April 2nd, 2023 at about 11:30 AM Eastern Time. The interview was done virtually over the phone. The interview included the prompted questions and other factors relating to project leadership, focusing greatly on the project leadership/experiences of the interviewee (Kenneth Roraback).

II. Background and Suitability of Interviewee Kenneth Roraback

Through the MIT Alumni Advisors Hub, I was able to connect with Kenneth Roraback. He graduated from MIT in 2006 with a degree in Physics and Theatre. In addition, he also received his Master of Fine Arts from the Parsons School of Design. His professional experience includes many different institutions, such as a former research assistant for the MIT Center for Space Research, web developer/designer at Kensho Technologies, Fast Orientation, and HomerLearning, Inc. Much of his work is in the realm of startups and smaller companies in the engineering discipline of web development. This is partly the reason I selected Kenneth as my personal engineer journey will likely lead me to web development after my Bachelors of Science and Masters of Engineering degree in computer science/web development. Kenneth is certainly a leader as he is the current Director of Product Design at Teleport with over a decade of startup leadership. He has led dozens of projects in different tech disciplines as well, such as network security, artificial intelligence, user authentication, etc. In short, Kenneth is clearly a suitable candidate for an engineering project leader. His credentials can also be attested through his 30+ consultations he has done through the MT Alumni Advisors Hub, where he regularly provides guidance on launching a startup, personal leadership, product development, and working in project teams for customer satisfaction.

III. Answers, Reflections, and Analysis of Interview Questions

A. *Please describe some of the engineering projects that you have led, and why they were successful or not.*

Kenneth described three different projects that comprised a large chunk of the interview. The first project that he led was at a child literacy startup called HOMER[1]. This startup has a goal of giving kids a reliable and stable start to their learning journey. In particular, this is an early learning program for literacy. A project that Kenneth described dealt with in the context of building a new web application was Decodable Readers. These are essentially simple readers or books that are designed to contain the specific grapheme–phoneme correspondences students have learned, thus providing learners with the opportunity to use their developing segmenting and blending skills to read words in order to develop automaticity, or the ability to recognize words quickly and effortlessly, and experience independent reading success[2]. The project involved building new web user interfaces for children to be able to record and play back certain words in order to reinforce best practices in literacy. Breaking down and sounding out words, instead of random guessing can be more effective, thus the project team wanted to build out a frontend and backend of the application, thus making frontend components, installing libraries, building out development infrastructure, writing API calls, configuring database queries, populating user profiles, handling events, running express server, etc. This project involved different subteams that were essentially responsible for the frontend, backend, testing, and integration along with some ad-hoc groups of engineers. Kenneth's role was in frontend development in trying to get a user interface implemented. This project turned out to be successful as Kenneth mentioned that there was an improvement of up to 73% in some cases of an increase in reading scores. The reason for the success that Kenneth provided was the amount of user testing and iteration that was done. In other words, there were varying levels of prototypes of the application that were tested on target audiences. Along with that, there was an outside expert on literacy brought in to help make a credible product implementation.

This project has numerous important takeaways personally. The most important one is about the need to test everything, essentially all the times that make sense. Without doing an iterative process, it is hard to deliver something of immense value. I will seek to make sure to always test my implementations as much as possible, consistent with Kenneth's experience, such as via glass box testing or black box testing. I also took Kenneth's advice about testing to be an invitation to involve more stakeholders in the process to have a diverse set of perspectives to make the product implementation robust and usable.

[1] https://learnwithhomer.com/
[2] https://www.readingrockets.org/article/what-are-decodable-books-and-why-are-they-important

Bringing in experts is likely a good decision most of the time if the deliverable is convoluted in some way. In this case, I think Kenneth's project team was great at involving key stakeholders, even doing research/getting into contact with the Assistant Secretary of the Education Department to hear the different feedback before the deliverable. Furthermore, I think Kenneth's story reinforces the need for cheap prototypes first before investing too many resources on A/B testing as once the engineering infrastructure is set, it is ineffective and cumbersome to change. Planning the development infrastructure and the end product infrastructure is key as the budget is quite low, thus having cheap infrastructure to test out before the real one can be cost-effective and performance-effective in terms of creating value. Lastly, Kenneth's project being successful can also be attributed I think to the distinction in the target audiences of the deliverable. Creating a tool for children is far different than creating a tool for average adults, thus the assumptions behind the project need to always be challenged, since there are nuances as to tech-savviness, familiarity, etc.

Kenneth's second project that he led was with Teleport[3] in building a small tool for displaying the appropriate access controls for developers/engineers of a project. This would allow for a faster and effective way to develop new tools as infrastructure changes can be detrimental if not organized well. Kenneth mentioned that there was much discussion of a certain type of authentication service, such as audit, okra, and other integrations. To move past this blocker, more testing was also done in order to create different, simpler, mock-up implementations of how to give an engineer access to a Kubernetes Cluster, connect to a Grafana instance, or add a node to the cluster. This deliverable tool was eventually a success due to the testing, feedback, and iteration process. I personally think that having variation of web tools in different environments/stages of development can be a decent way of ensuring projects are on track to success. Linking these to project management tools can also mean that the entire team can be up to date on smaller issues that arise due to iteration. Lastly, I think that screenshots and non-backend heavy tools can be readily accessible to spin out quick mock implementations to preserve budget and time, but still testing impact of value within a project team.

Kenneth also briefly described a third project that was not successful, different from the first 2 projects. This was at the network security startup that he worked at, where the goal was to implement methods or procedures to prevent harm in the case of a network outage, such as effective caching, error handling notices, and automated response sets. The project was not successful because the startup was in its early stage(seed funding) and project engineering leadership was effectively not present at all it appears. For instance, there was no concrete way to collaborate with the codebase among multiple individuals

[3] https://goteleport.com/

and no strong code review process. Without feedback, this development process took a big hit on the entire project/essence of the company as there were many key performance indicators tied to this project. Kenneth also mentioned that there were also many other engineers who were starting in their own early careers, thus fairly inexperienced as to how to actually implement a product in a project team. This ultimately meant that the company was weighed down heavily by a combination of inexperience, uncertainty, and ineffective development environment. Knowing this, Kenneth decided to help implement some changes for the project team, including an effective GitHub code review process and an entire restructuring of the company's tech stack/engineering development structure. This took a fair amount of time, but better organization of files, codebase, documentation, and shared expectation of development allowed the company to make great strides. From a project leadership perspective, agile process with weekly planning meeting, grooming tasks meeting, estimation meetings, discussions, and simply better structured communications (i.e. Developing and Deploying Structured Communications) made the projects attainable and more effective. Even though it was a fairly smooth project team process, the startup did not end up securing enough funds to continue because the poor early architectural decisions were too costly to overcome. From this story, I resonated with Kenneth's proposition to always be willing to take a risk earlier rather than later if the circumstances are clearly so off that a project team cannot create enough value to sustain itself forward. I also think that a lack of organization or the feeling of organization can make anyone feel confused and not be willing to work together as a team. Even though Kenneth did not mention this explicitly, I think he would also agree that too much planning can be a hindrance to the value-creation process of a project team since not enough precious energy is being put to creative uses.

B. *Describe what has made you effective as an engineering project leader.*

Kenneth mentioned a few key reasons, including project tracking software, project roadmap, visualizations, and prioritization. He mentioned that it was vital to display all of the known information in a readable format to all of the relevant team members in a project. This can help see where the vision and priorities of the project are. A project management tool like Jira or GitHub issues can help out with this transparency and highlight where the potential blockers can be before they even materialize. Furthermore, Kenneth said that there is always more work than once can possible do, thus explaining any trade offs or prioritization to everyone keeps the same level of expectation intact. Vision and the goals of the project must also be communicated easily to the team in order for there to be complete buy-in from the team in terms of working together to create value despite the range of constraints that also should be documented well. I personally think some central location is a great idea for most projects or complicated life things. For instance, from a personal academic perspective, it is very frustrating when some

materials for some classes are outside of a course management system, that is, the materials are distributed rather than centralized, which makes finding them harder. This can be coupled with a busy schedule to create a lack of trust and loyalty (i.e. Fostering Trust, Loyalty, and Team-Building) in the team. I think that I will continue to be as transparent as possible and unless there is a strong reason to not do so, I will just publish a link to all of the meeting notes, drive links, etc. where anyone can see everything. I will also make sure people know about this since this transparency is not known to people, thus unnecessarily eroding trust. I think Kenneth also implies that in general, it is good to foster trust not only within teams, but also across teams. For instance, there is an inherent relationship between frontend and backend teams, thus it is important to make sure there is clear communication and documentation between the teams for a smooth project implementation process.

C. *What are your strengths and weaknesses as an engineering project leader? Do you have a formal or informal plan to strengthen your weaknesses? How have you overcome past weaknesses?*

Kenneth described his weakness of not being confident enough sometimes, especially in the earlier parts of his career or at early-stage startups. He mentioned that he has been involved in a variety of situations with varying degrees of good project engineering leadership, from basically no organization to very fine-grained organization. For the no organization case, he wanted to not intervene much as to avoid conflict or to maintain the status quo/have sympathy since he did not have the repertoire with other team members. I can also relate to this weakness as I often do not like to cause trouble for minor issues when the removal of a few minor issues would equal the removal of a large issue. In addition, I also don't like to interrupt people or put forth my perspectives all out on the floor immediately. To this effect, Kenneth mentioned that he had an informal plan of simply being involved with more responsibilities, especially volunteer opportunities, in order to give him the most experience he can get. I usually like to do this as well since I have tried to dabble in many many groups/organizations at MIT from research to cultural to religious to recreational to greek life to leadership to sports to classes to traveling. These experiences together are helping me be assertive when needed since I do believe that not talking often speaks volumes. Kenneth also had a somewhat formal plan to seek out his mentors/other close team members and learn to make decisions with certainty. I also have learned that without concrete decisions, many teams simply keep losing focus in team meetings, etc., thus it is important to bring everyone back and actually make a decision sooner rather than later. Kenneth's biggest strength then became experience instead of inexperience and also becoming really good at his part of the job in relation to the overall project. By doing his work with extreme precision and high performance, he was able to exude confidence and respect, thus making the flow of the team better due to

high expectations. Kenneth also became an expert at estimating realistic project tasks in order to not undershoot or overshoot all the time, since that can kill team morale at times. I personally think his strength is something that I want to work on as well since estimating other people's abilities and assigning tasks/managing a team can be sometimes unnerving. I also think that Kenneth would also agree that I should probably get better at time scope as to the fact that decisions that are made early can and do often have an impact later down the road even if it is hard to see in the moment.

D. *What are your recommendations/advice/guidelines to engineers who are leading, or about to lead, an engineering project?*

Kenneth's main advice/guidelines have already been answered/are weaved throughout the previous answers, but ultimately include 1) realize the scope of the project early on, 2) realize that early team decisions impact lasts a while, 3) tangle with the tradeoff between shortcuts vs long-term planning, 4) figure out your leadership style, and 5) read Radical Candor[4] in order to grasp the framework of feedback in projects. I think the feedback framework is really interesting as I sometimes like to avoid feedback as it takes effort/time away from the present task at hand. However, I think Kenneth would say that without that feedback, the projects he has worked on for many startups would all be crumbling away and no innovation would be produced, thus this is a good recommendation. In addition, I think I would apply his advice on some web development tools since I often complete a basic MVP first under a few simple assumptions and then ask for feedback from a supervisor, but should probably confirm instead of assume.

IV. Key Engineering Project Leadership Guidelines

1) Keep funding in mind and try to be proactive at saving money when possible
2) Don't reinvent the wheel for what works, both in terms of project management and technical implementation
3) Propose slightly optimistic timeframe, budget, scope, etc. and work with everyone to make it realistic, taking into account the buy-in of the team
4) Provide all the known information about the project in one centralized location and send out concise info reminders to remind everyone about existence of transparency
5) Provide testimonials or some realistic scenarios for decisions about project management and technical implementation to bring home the point of the "why" for a particular decision
6) Focus on one major blocker at a time unless the team feels ready/has enough trust, loyalty, etc. to take on multiple challenges

[4] https://www.radicalcandor.com/

7) Have a clear start of a project, such as a big “Go” button to make sure everyone knows what’s happening on the highest of levels

V. GEL Capabilities of Effective Engineering Leaders and Engineering Project Leadership

There are many GEL capabilities that are embedded into the startup-heavy project leadership experience of Kenneth and the interview itself. For instance, Upholding Responsibility, Sense of Urgency, and Will to Deliver was key for Kenneth to be able to see out a feature of the child literacy Decodable Readers tool from ideation to deployment. This involved critical responsibility due to the target audience of children and the inherent risks involved. Furthermore, robust testing means there is an expectation that the tool will deliver to perfection or without too many risks. Exercising Resourcefulness, Flexibility, and Resilience was also permeated throughout the interview since it goes hand in hand with testing, feedback, and iteration processes because changes are needed in the tools. I personally gained a new sense of the capability of Fostering Trust, Loyalty, and Team-Building since Kenneth expressed that being confident and decisive is a form of trust. He shared a story about how a team member came to him with a concern that could not have been addressed since the project was too far along to change. Instead, Kenneth wrote a 4-paragraph letter agreeing and offering gratitude for the concerns and clearly explaining what the lay of the land is before meeting with the concerned individual. Active listening is Fostering Trust, Loyalty, and Team-Building. Being decisive is Fostering Trust, Loyalty, and Team-Building. Team trust, testing, and resiliency were common themes throughout the interview in relation to the GEL program/capabilities.

VI. Personal Path Forward

I personally will try to converse more with the team and try to establish some expectation early as without a solid foundation, I think a project team could be on the track to being doomed from the beginning. I will try to make a series of incremental changes and get feedback instead of making assumptions because I do not want to bother other people. I will also try to go ahead and brainstorm a plan or even do a mock implementation(in web development context, try to utilize a local server) of a testing paradigm(with the team, construct a baseline set of steps/criteria) and what the end deliverable could conceivably look like(invest in getting outside experts as needed). I also will use more project management/development tools(GitHub development framework or a centralized system) that exist out there to get the basic functionality that is not “controversial” out of the way

before having more discussions(having some routine) about deeper issues that relate to the ultimate deliverable to be used by consumers. I will also try to be consistent with providing concise and concrete decisions(make it clear what decisions should be made when) instead of wishy-washy assumptions or just asking questions all the time unless it is needed. Overall, expecting changes(thus having a buffer on time/scope on testing and implementation processes) and talking more with the team at hand(questions platform) would be some ways of utilizing Kenneth's guidelines.