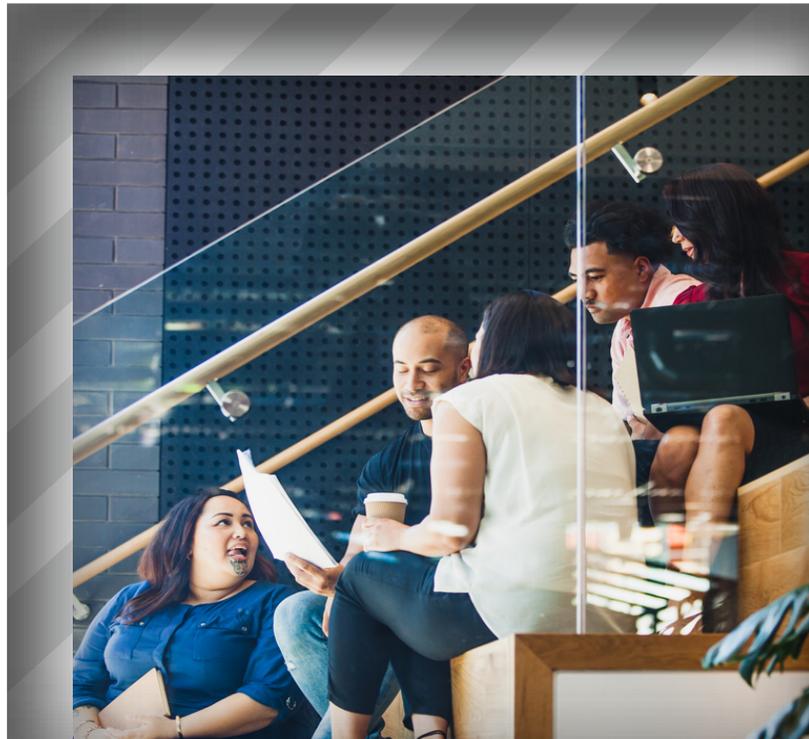




3 Simple Things That Could Make or Break Your Development Team



Joshua Candamo, Ph.D
Director of SDK Development





It doesn't matter if you are a small startup or an enterprise giant, or if you have in-house development teams or contractors. The effectiveness of your tech teams is an integral part of your business success and strategic growth.

We live in a world driven by technology, and technology is changing fast. Companies can't escape this reality. It's either evolve with technology or become extinct. Don't take my word for it. Think about the evolution of technology in industries like transportation (Uber), retail (Amazon), and video (Netflix). You can try to escape reality, but you will probably fail.

One of the first things that comes to mind when talking about software development teams is to ask if teams are absolutely necessary. Can't we rely on individual tech professionals instead of teams working for our companies? Maybe the whole is not more than the sum of its parts?

The fact is that, in general, teams outperform individuals. When people work in a team toward a common goal, they combine their skills. In a team, individual [performance increases](#), and people are able to solve more complex problems, efficiently and effectively.

My name is Joshua Candamo. I'm a technology leader with a PhD in computer science. My background is pretty diverse, and includes considerable experience programming as well as over 14 years of technology leadership.

I am currently Director of SDKs for Accusoft, a software development company specializing in content processing, conversion, and automation solutions. My engineering group collaborates with about 40 people including in-house software developers, offshore contractors, technical writers, product management, quality, marketing, and sales professionals.



I want to share what I've learned from my personal experience of building development teams over the last 14 years, and a few useful tips to doing so successfully.

Without further ado, let's talk about the three simple things that I found can make or break development teams.

To get started, let's point out the obvious. Don't fight nature; embrace it.

If you try to plant a rose in the middle of the desert, it will most certainly die.

You can't fight nature. However, if you understand nature, you can embrace it and make decisions that align with it.

You can simply build a greenhouse in a harsh environment and succeed at growing a rose pretty much anywhere. Using the same logic, there are some foundational pieces that you have to anticipate in order to build a successful team. Avoiding basic considerations of team building will likely make your development team fail or underperform.

Team building is a broad and complex topic. And, it's also a topic that I'm passionate about. Not everything around team building is complicated. However, most initiatives require a methodical approach to correctly execute them.



#1 - Team Size: Bigger Is Not Better

Over the last decade, technology giants like Google spent millions funding behavioral studies, like [Project Aristotle](#), in order to understand what makes great teams great. Google would tell you that in the secret sauce of great teams, “communication is the key.”

In the best teams, members listen to one another and show sensitivity to each other's feelings and needs. These are simple considerations that you would expect from normal social behavior.

In my article, [“Tips for Team Building: Keep Size Small and Encourage Gossiping.”](#) I discuss the importance of healthy communication for team building. I also explain why smaller teams tend to have better communication, and better communication often leads to better teams that are highly productive.

Psychology studies have found that there is a biological limit ([a direct relationship to the neocortex size](#)) to the number of stable relationships a person can maintain in their lives. Putting it simply, your brain is limited to having social and meaningful contact with only a handful of people at each point in your life.

Considering Google's findings and our natural social limitations, it makes sense to limit the number of people on a team to a level that is conducive to building significant relationships within it.

Smaller teams make sense from a biological standpoint because they are ideal platforms to allow members to build strong relationships.

“Smaller teams make sense from a biological standpoint because they are ideal platforms to allow members to build strong relationships.”

Let me caution you. There is no such thing as a universal and unequivocal perfect team size. However, team size and the advantages of small teams are well studied.



If we ask Amazon's CTO, Werner Vogels, he will give you an interesting and "fairly precise" [answer](#). Your team should be small enough to be fed with two pizzas. For more pragmatic advice, I recommend that you limit your team size to the smallest number of people necessary to accomplish your projects.

“Limit team size to the smallest number of people necessary to accomplish your projects.”

If at this point I ask 10 people reading this article, "What questions do you have so far?" I bet 9 out of 10 people will ask:

What team size do you recommend?

While the research on optimal team numbers is not conclusive, researchers tend to agree with a range of 5 to 9 people.

In my experience, keeping teams between 4 to 6 people is the right place to start. Clearly, depending on the complexity of the project, this could vary. However, if you find your team is larger than six people, ask yourself if you have too many people or too much redundancy in a given skillset.

Having a great team depends on much more than having the optimal size. Given the



complexity of human nature, there are many other variables that play a part.

Next, I'll go over another piece of advice I found can make or break a team in terms of productivity; and it's also easy to implement without requiring any capital investment.

#2 - Team Composition: Art or Science?

It is well known that diversity [increases team productivity](#). But... What is team diversity? And even more importantly... What is diversity in the context of development teams?

In general, diversity has to do with encompassing a wide range of experiences, backgrounds, and interests.

When discussing diversity, the most common aspects we think about are gender, ethnicity, and religious beliefs. When talking about development teams, diversity also has to do with many other factors like work experience, personality, and technical skills.

As I mentioned earlier, in general a team is greater than the sum of its parts. A diverse team is even greater.

“A team is greater than the sum of its parts. A diverse team is even greater.”

Diverse teams are more likely to make better decisions than either individual team members on their own, or other teams consisting of people with more homogeneous skill sets and experiences.



A team's performance and decision making capabilities have to do with what is known as its [collective intelligence](#). Collective intelligence is the “*group intelligence*” that emerges from the collaboration of a group of individuals.

Interestingly enough, collective intelligence is not strongly correlated with the average or maximum intelligence of individual team members. In other words, adding really smart people to a team doesn't necessarily make the team as a whole any smarter, or any more likely to make better decisions.

“ Adding really smart people to a team doesn't necessarily make the team as a whole any smarter, or any more likely to make better decisions. ”

What makes a team smarter? More importantly, what makes a team capable of making better decisions? This is a topic that [Google set out to answer](#) by analyzing more than 200 interviews with Google employees and looking at more than 250 attributes of over 180 active Google teams. Google found that the top team dynamic that sets successful teams apart is [Psychological Safety](#). Psychological safety has to do with team members feeling safe to take risks and be vulnerable in front of each other.

Google's finding is no surprise when you look at related research in behavioral psychology, which indicates that there is a significant correlation between collective intelligence and the

average social sensitivity of group members.

At this point, you are probably asking yourself...

What is the ideal team composition?

Let's say you start from my previous advice and limit a team's size to the smallest number of people necessary to accomplish a given project.

Most of the work that development teams have to deliver is complex in nature. Since educational and age diversity has been linked to [positive team performance](#) (at least when the need for cognition is high), my best advice is to try to have at least one junior and one senior team member on each team.

If your organization is following some kind of Agile software development methodology, like Scrum or Kanban, you also want to ensure that the team is not composed of individuals with unique skill sets. For example, one single person who can design APIs and one who can write automated tests.

It is natural, and expected, to have some skill specialization within a team (e.g., a front-end or a QA automation expert), as long as there is some kind of skill redundancy spread within the team.

Skill redundancy prevents bottlenecks when any team member becomes unavailable; it also incentivizes collaborative knowledge. Collaborative knowledge is particularly interesting, since it has been linked to corporations gaining an edge in terms of creativity and innovation.

“Every team should have at least one junior and one senior team member, and enough skill redundancy to avoid bottlenecks when a team member is unavailable.”

My last advice covers a fundamental, and often overlooked, component of team building.

#3 - Team Voice: Actions Are Louder Than Words

Communication is key for team building and vital to the development of a healthy corporate culture. From a team building perspective, there are few things more important than encouraging communication by providing teams with the right platforms for doing so.

There are two basic types of communication: internal and external.

Internal communication takes place among members of a team, and external communication is between the team and stakeholders or outside members of the overarching organization.



In my article [Innovation Through Social Re-Engineering](#), I discuss the importance of internal team communication and social interactions in order to foster a strong, collaborative teamwork environment.

How can you give your teams a voice?

Giving technical teams a voice allows them to provide feedback about the corporate environment and technical leadership structure.

A simple, cost-effective way to request feedback is to conduct anonymous employee surveys.

As business leaders, we crave feedback and sources of information that can be actionable. However, in order to receive concise feedback, as Ed Catmull recalled in his book [Creativity Inc.](#), you have to ask.

Generic feedback mechanisms, like an open door policy or asking for general feedback during one-on-one meetings, are not effective ways to gather concise, valuable feedback.

It is considerably different to ask, "Do you have any feedback for me?" rather than being more specific and asking: "What do you think about the current goals for this quarter? Or the way we are measuring your progress?"

Most of the time, vague questions will result in vague answers. Being specific makes it easier for employees to provide more valuable feedback.

Anonymous surveys also offer a perfect platform to get concise feedback for multiple areas of concern, in a safe environment for the employees.

“ Anonymous employee surveys are low cost, simple to implement, and offer an ideal platform to request specific feedback for multiple areas of concern. ”



Next, let's discuss some employee survey dos and don'ts.

To start, be mindful that it can be uncomfortable for employees to provide feedback if their opinions include constructive criticism, especially feedback about management.

There could be some uncertainty about the degree of anonymity of a survey, which is of special concern for surveys with sensitive questions. Typically, respondents' concerns will translate into lower participation.

The first time we sent an employee satisfaction survey to my current engineering product group at Accusoft, the response rate was 50%. However, follow-up meetings to communicate action plans based on some of these responses was critical to getting 100% response rate in the next two surveys.

“After each team survey, there must be a follow-up meeting or communication to the team disclosing an action plan to close the feedback loop.”

The basic idea of a team survey is simple. The team provides feedback, and the company responds. Why? Because the company cares about their employees and their concerns.

The relationship between the leadership, or the business, and a development team is just like

any other relationship. It takes two to tango.

When crafting a response to close a survey feedback loop, avoid feeling the necessity to answer all concerns or questions. Don't have a manager address the survey responses during a public 3-hour meeting.

Focus on patterns. Focus on areas that are of strategic importance to developing your corporate culture or vision. Don't be distracted by minutiae or bitterness. Stay positive.

Never use feedback against employees, or try to reverse engineer the feedback to find the respondent's identity.

Why? Because you care about the feedback. Good and bad.

Ensuring the survey's anonymity contributes to a culture of trust and encourages honest responses without fear of retaliation from management.

And last, but not least, a survey is about listening and providing teams with a voice and a healthy outlet to provide feedback or constructive criticism. A survey is not about you, or the company. Again, it's about listening.

Sometimes, feedback might not be fair or useful. And that's ok. Receiving feedback that is not constructive can be frustrating. However, from the communication standpoint, you've already succeeded.

“When closing the feedback loop for a survey, avoid feeling the necessity to answer all responses. Focus on responses that align well with strategic areas of importance to your corporate culture and vision.”

At this point you are probably saying, I get it. Surveys are easy to do and cost effective.

What questions do I ask?

It depends.

Questions you ask in a survey should align well with your strategic vision and corporate culture. Ask questions that you really need answered!

Are you struggling with turnover? Ask for feedback or suggestions about how to improve work-life balance and overall satisfaction.

Every company has different struggles and goals.

As an example, here is the last employee satisfaction survey for our SDK engineering product group at Accusoft:

Are you a contractor or employee?

- 1 - Contractor
- 2 - Employee

I'm satisfied with my role on my team.

- 1 - Strongly disagree
- 2 - Disagree
- 3 - Neither agree nor disagree
- 4 - Agree
- 5 - Strongly agree

I'm satisfied with the work-life balance I have in the SDK group.

- 1 - Strongly disagree
- 2 - Disagree
- 3 - Neither agree nor disagree
- 4 - Agree
- 5 - Strongly agree

I'm satisfied with the technical training provided in the SDK group.

- 1 - Strongly disagree
- 2 - Disagree
- 3 - Neither agree nor disagree
- 4 - Agree
- 5 - Strongly agree

I am a better software developer than I was a year ago.

- 1 - Strongly disagree
- 2 - Disagree
- 3 - Neither agree nor disagree
- 4 - Agree
- 5 - Strongly agree

Compared to one year ago, we are doing better "holistically" as a product group.

- 1 - Strongly disagree
- 2 - Disagree
- 3 - Neither agree nor disagree
- 4 - Agree
- 5 - Strongly agree
- 6 - I wasn't here a year ago

Members of my team value and respect each other's contributions.

- 1 - Strongly disagree
- 2 - Disagree
- 3 - Neither agree nor disagree
- 4 - Agree
- 5 - Strongly agree

I feel safe openly discussing any issue with my manager, including providing constructive criticism.

- 1 - Strongly disagree
- 2 - Disagree

3 - Neither agree nor disagree

4 - Agree

5 - Strongly agree

I'm satisfied with the SDK group and its culture.

1 - Strongly disagree

2 - Disagree

3 - Neither agree nor disagree

4 - Agree

5 - Strongly agree

What do you like best about the SDK group and its culture?

What would you change about the SDK group and its culture?

I have a clear understanding of the vision and goals of the SDK group.

1 - Strongly disagree

2 - Disagree

3 - Neither agree nor disagree

4 - Agree

5 - Strongly agree

What would help you better understand our vision?

Is there something else you think we should have asked (or skipped asking) you in this survey?

Tell us more. (Do you have ideas, suggestions, comments, complaints, praises?)

The Agile Approach to Team Building

Building the right development team is one of the most important endeavors for organizations today. The odds that your organization is going to "get it right" from the beginning are low. But, what is "right" in an Agile world anyway?

The right approach to building, or improving, development teams is to target a well thought out strategic direction towards team building. After all, your most valuable intellectual property has nothing to do with patents or technology, and everything to do with your people.

“Your most valuable intellectual property has nothing to do with patents or technology, and everything to do with your people.”

In general, dealing with people is not simple. In fact, nothing about behavioral psychology or team dynamics really is. Why? Because human nature is not simple.

When creating new or trying to improve existing development teams, you should take a methodical approach.



First think about the amount of the people per development team. Think along the lines of the minimal set of people needed to do all the work necessary to deliver comprehensive value.

Second, think about the team composition. A team should have enough diversity as well as expertise to be effective and avoid bottlenecks when a team member is unavailable.

And finally, ensure that the team has a feedback channel to provide information to the leadership so that you can improve moving forward.

Analyze feedback on a regular basis, as well as productivity data, and don't be afraid to make small changes and experiment to challenge the status quo and introduce incremental improvements based on observations. Sounds Agile? Because it is.