Collaborator

Building a Development Culture of Collaboration

SMARTBEAR

Content

Introduction 3

Trust 6

Clear Expectations 10

Collective Ownership 16

Communication Channels 18

Fast Feedback 20

A Collaborative Development Culture 22

Citations 24

When 1 + 1 = 3

Collaboration is the promise that when people work together, they are better than the sum of their parts. They are able to be more creative, communicate more effectively, and fulfill their full potential.

But what does it truly mean to collaborate? And how can a team set itself up for success? In this eBook, we will be outlining the keys to collaboration and tactics that your team can adopt to truly excel.

Introduction

Take a look at the ants on the front cover. Ants are all about collaboration and team work. They bridge gaps by collaborating. Now, they will also sacrifice themselves for the good of the colony. Some species, when under attack, will grab the intruder and then secrete some chemical that causes them to explode, killing themselves and the intruder. All to protect the colony.

Hopefully, your office doesn't work quite like that, but still you can appreciate the power of teamwork. When a development culture is intentionally collaborative, it greatly improves people's lives and allows all of us to build quality software, which in turn positively impacts our businesses and our customers.

Dr. Steven Covey stated, "When people begin to interact together genuinely, and they're open to each other's influence, they begin to gain new insight."[1]. While the execution might often be lacking, companies understand the importance of this.

In the 2017 Human Capital Trends Report by Deloitte Insights, 94% of the companies surveyed see agility and collaboration as critical to the organization's success.[II] 32% are actively working to make their organization more adaptable and team focused. So what is the goal?

Defining a Culture of Collaboration

On its own, a culture is defined as the attitudes and behavior characteristics of a particular social group. Collaboration is the act of working with someone to produce or create something.

So ultimately, a collaborative culture is:

> **So ultimately, a collaborative culture is:** a social group whose attitudes and characteristics allow them to work together to produce or create something.

Cultures of collaboration really stand out when the pressure is high and your team needs to execute. A great example of is how NASA responded during the Apollo 13 mission. Two days after the launch of their lunar-bound ship, an oxygen tank for the Service Module exploded. As a result, carbon dioxide began building up in the Command Module.[III] The three astronauts were stuck in space, running out of oxygen, and could only communicate with engineers on the ground via a radio – they couldn't see each other.

Apollo 13 Mailbox at Mission Control. (1970, April 15). NASA. Public Domain

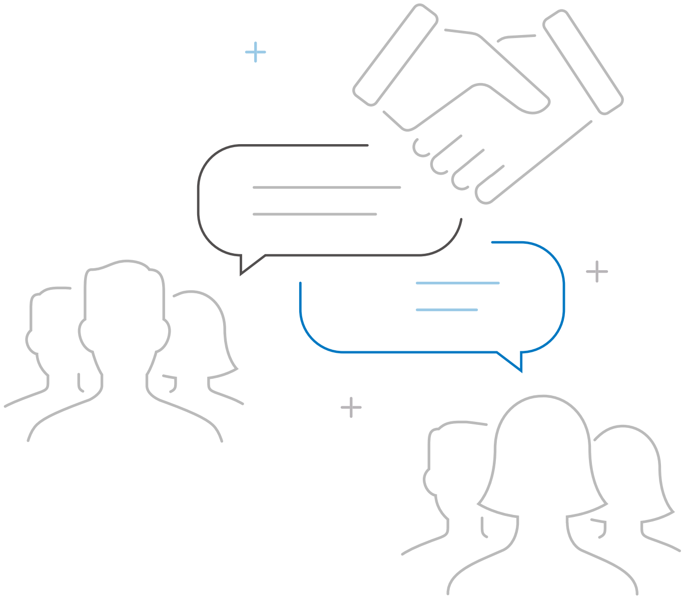
The mission control team, though incredibly bright, didn't have a solution ready for this urgent problem. They called the engineers to gather up the miscellaneous parts that would be onboard on the ship and make something work. In this picture, you can see the makeshift prototype that would need to be assembled onboard in order for the astronauts to have breathable air. Referred to as "the mailbox", this contraption brought together a cubed-shaped canister with a cylindrical canister-socket through a return hose. Not necessarily the most intuitive device.

Once they had the design, mission control had to convey how to build this part to astronauts thousands of miles away over a radio before they passed out from asphyxiation. No diagrams, no video, but they got it done. While the mission didn't accomplish its intended goals, it illustrated why collaboration and communication are so critical to mission success.

The Collaboration Hierarchy

To achieve this level of collaboration, teams need to embed it in their culture over time. For this ebook, we will be looking at collaboration with the same framing as Maslow Hierarchy of Needs.[IV] What are the fundamental components to build your culture around and where do you go next? For each level of the hierarchy, we will provide constructive tactics that will help your team collaborate more effectively.

The first part of forming a solid team is trust.



Trust

A Gallup poll in 2012 found that when employees trust their leaders, their engagement at work is better than 1 in 2.[V] Inversely, when they don't trust their leaders at work, engagement goes to 1 in 12! Trust is a short word, but it encompasses whether or not a team has a workable foundation.

These quotes add some more context to what trust really means for teams:

> "
>
> *Trust is maintained when values and beliefs are actively managed. If companies do not actively work to keep clarity, discipline and consistency in balance, then trust starts to break down.*[VI]
>
> **Simon Sinek**

> "
>
> *When people honor each other, there is a trust established that leads to synergy, interdependence, and deep respect. Both parties make decisions and choices based on what is right, what is best, what is valued most highly.*[VII]
>
> **Blaine Lee,**
> **author of The Power Principle: Influence with Honor**

"

If you don't have trust inside your company, then you can't transfer it to your customers.[VIII]

Roger Staubach, Quarterback of the Dallas Cowboys

So, think about it. Do you 100% trust your coworkers and your boss? What about your management team?

If trust isn't woven into your development culture, many people may feel like they are in the Hunger Games, only looking over their shoulder and being defensive. At a minimum, some teammates may be afraid to be vulnerable and truly let down their guard.

Maybe they don't feel as smart as their coworker. Maybe their coworker took credit for something they did. Maybe they are a contractor and don't feel like their position is secure. This could be for any number of reasons.

So how do you actually build trust?

1. Promote positive, open communication
2. Treat others with respect and courtesy
3. Give trust freely and assume you're not under attack

A good example of giving trust freely is in a relationship, when your significant other says something to you that makes you feel like you're under attack. Take a second and assume that they don't have malice towards you. Whatever they just said isn't an attack.

Of course, there are going to be times when they are actually attacking you, but that's likely not the majority of the time.

If you can promote positive communication, treat others with respect and give trust freely, then you will be fostering a free mental enterprise.

A Free Mental Enterprise

A free mental enterprise is a mental economy that can produce exponentially when given an open collaborative forum for ideas, fully unleashed and without judgement.

There are two critical ways to encourage free mental enterprise. They are active listening and asking quality questions.

Active Listening

Active listening builds trust because it allows for team members to feel heard while also sharing information on what they feel and think about a topic or situation.

Whether it is during a standup or sprint retrospective, if you are making an intentional effort to listen to concerns and make room for those concerns to be aired, it allows for open communication across the team.

Making room for communication means withholding outright judgment on comments, which can stifle honest feedback.

It also means reducing distractions. For example, when everyone has their laptop open or when everyone is looking at their cell phone on a meeting, do you believe that the person speaking feels that they are being listened to?

Not only is it important to actively listen, it is important to remember that there are cultural differences between employees and English isn't always the native language.

Active listening is a practice that shows respect and works toward building trust between two people.

The flip side of active listening is speaking, but rather than telling people how or what to do, or what you think, you might get further with quality questions.

Asking Quality Questions

Which of these questions below will encourage a quality answer?

- "Are you sure that is the best way to implement the algorithm?"
- "What was your process for deciding on how to implement the algorithm?"

It's the second question, but why?

The first question allows the person to just respond with "Yes" or "No". No further conversation necessarily. The second framing encourages the individual to reflect on why they did what they did. It allows them to fully explore their work and then articulate their thought process to the person asking the question.

In *The Coach Model* by Keith Webb, he describes 7 different types of questions [IX]:

1. Closed questions – Do you agree with this logic?
2. Open questions – What is happening in this method?
3. Directive questions – When will you take the time to finish fixing this defect?
4. Emotion questions – how do you feel about the current project you're working on?
5. Fact questions – Who are the members of this team?
6. Why questions – What factors did you take into account before you wrote this program?
7. Permission questions – Do you mind if I ask you a question?

It can be helpful just to be self-aware about how we are framing and potentially limiting communication by using too many closed questions when trying to get to the truth.

Charles Price from the People's Church in Toronto once told a story about an old man that was walking along the side of the road with his dog, and his mule.[x] A truck came flying around the corner and knocked the man, his mule and his dog into a ditch. The old man decided to sue the driver of the truck for damages.

While the old man was on the stand, the counsel of the defense cross examined the man and asked him this question. "Didn't you tell the driver of the truck, at the time of the accident, that you were perfectly fine?"

The man responded by saying, "I was walking down the road with my dog and my mule..." The attorney cut him off and said, "It's a yes-or-no question. I'll ask you again. Didn't you say at the time of the accident that you were perfectly fine?"

The man looked up and said, "I was walking down the road with my dog and my mule."

"Stop, stop, stop!", responded the attorney. "Your honor, would you please insist that this old man simply answers my yes-or-no question."

The Judge said, "He obviously has something to say, I'll let him continue."

So, the old man starts again, "I was walking along the side of the road with my dog and my mule when that man drove around the corner too fast and knocked us into the ditch. He got out of his truck and saw that my dog was in pretty rough shape. He went back to his truck, got his rifle, and shot my dog. He then noticed that my mule had two broken legs and he shot it. Then he looked at me and said, "How are you?" I told him, "I'm perfectly fine."

Quality questions build trust, allow for open

communication, and convey respect because they indicate that you are trying to better understand their position and are seeking additional clarity.

Quality questions allow for free mental enterprise

because they deepen our understanding and invite new information to broaden our understanding of a problem or situation.

Clear Expectations

Once a team has a foundation of trust, the next consideration is setting clear expectations.

Do you know what is expected from you at work? What you are supposed to do most days?

Most of us understand what we are essentially supposed to do at work. We were hired to do a specific function that has a set of job descriptions to go along with it. We generally have a pretty good idea what we are there do.

Beyond that, are managers setting clear expectations for the group?

This is especially critical when your team is changing or when things are changing around your team.

One of our development teams at SmartBear recently switched to scrum. They do sprint planning and base everything on how many story points can be completed in any given sprint. So, as a team, they determine which tickets to pull into the sprint. Previously, they were assigned specific tickets, and that still happens now, but many would finish all of their assigned tickets and say, "I don't have anything to do."

In the case of scrum, we all need to be moving the football forward together. That means there are other tickets to work on, or developers that might need help, or testing that needs to be done so that we successfully finish the sprint.

The team is just getting used to having a new set of expectations and what it looks like to move forward together. This means that these expectations have to be reinforced constantly. At first, just about everyone was against it. Now, the majority appreciate the change. We started with one really large team by scrum standards and ended up splitting into two teams.

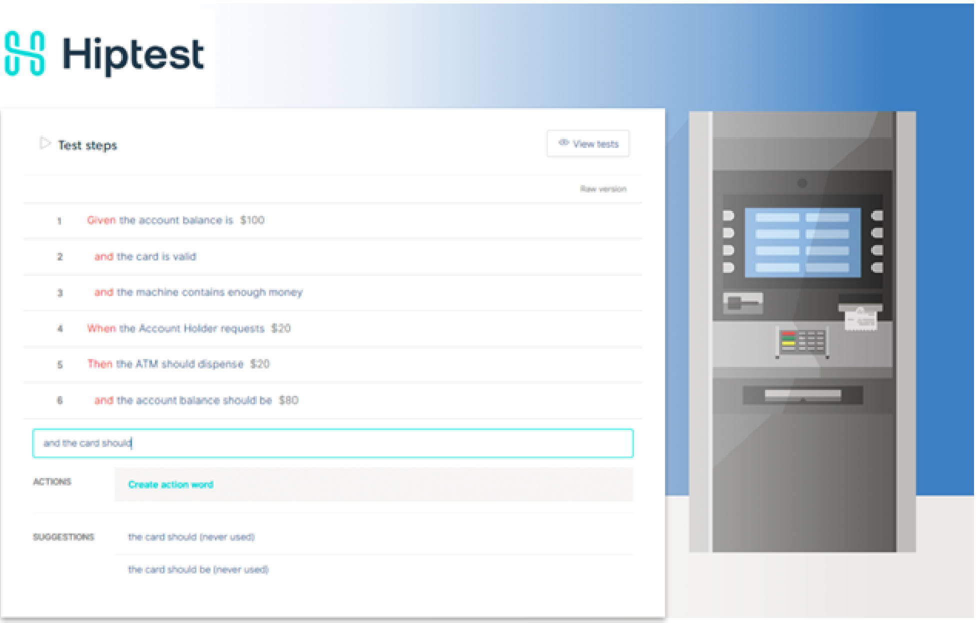
The first 5 sprint burndown charts looked like an EKG, but it is getting better. The team is starting to rally around their items and take ownership for completing everything. The daily standups and sprint retrospectives have been critical components to the team's growth during the transition.

It is extremely important that individuals and the team as a whole have clear expectations. It allows for everyone to be held accountable.

Setting a Clear Definition-of-Done

In addition to setting cultural expectations within the team, they are also thinking more about behavior-driven development and agile test management. They are using HipTest, a SmartBear tool which utilizes Gherkin syntax to help all members of the team, including non-programmers, define the business needs as part of the Definition-of-Done. This aids the developers while they are implementing the feature and provides clearer expectations around the Definition-of-Done.

As can be seen in this simple example, you define the Feature and scenario, using Given, When, Then statements as the syntax.



1 Given the account balance is $100
2 and the card is valid
3 and the machine contains enough money
4 When the Account Holder requests $20
5 Then the ATM should dispense $20
6 and the account balance should be $80

and the card should

the card should (never used)
the card should be (never used)

So, you have an Account holder that wants to withdraw cash and has sufficient funds:

Given they have a sufficient balance and their card is valid and the machine actually has money in it, When the account holder request 20 bucks, Then the ATM should give them $20 and the balance should be decreased and the card should be returned.

By plugging our scenarios into HipTest, we can then use that data to drive our test automation.

Building a Peer Review Process

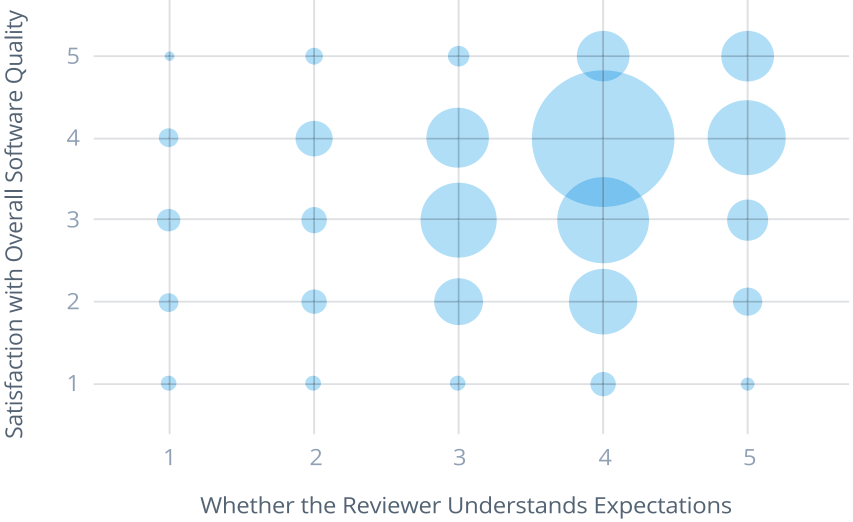
A clear Definition-of-Done is important for development and QA teams writing code and creating tests, but it is also important for the people that are doing your peer reviews.

In peer reviews, there are a few common roles. An author will create a review and then invite reviewers. A scrum master may want to serve as a moderator so they can keep track of where a review stands. If someone is new to a team, they might be just be an observer on reviews.

By breaking reviews into roles, you can then set explicit expectations for each. For example, you could say that authors need to take the time to first annotate the review and explain why they made the changes that they did. This expectation might result in the author catching their own coding mistakes. For reviewers, you might want to set an expectation on when reviews should be completed after assignment. Consider using review checklists to ensure that each review has completed explicit steps before being marked done.

We recently ran a survey titled The State of Code Review. [XI] The respondents that stated that they knew what was expected of them when reviewing code and were more likely to be satisfied with the overall quality of their software than those that didn't.

When the management team works with developers to set clear expectations for each role within the peer review process, teams find more defects and knowledge transfer occurs – which means you are raising up talent while improving quality.



n = 847

Raising Up Talent

One of our customers decided that he was going to personally conduct every peer review for his team, over a 9-month period. His boss thought he was crazy, but agreed to let him try it. It was incredibly successful. Not because they improved quality or were able to capture metrics, but because of how much his team improved.

In every review, he gave specific examples of how to do things. Not because they were his opinions, but in reference to best practice books on coding and style. He didn't always tell them the specifics on how to address something but instead gave them a pointer to the material and had them go read the resources on their own. His group ended up becoming the most successful group, from a financial perspective, in the entire company. It's an amazing testament to the importance of developing people.

He set expectations for how reviews needed to be conducted and then coached the entire team on it.

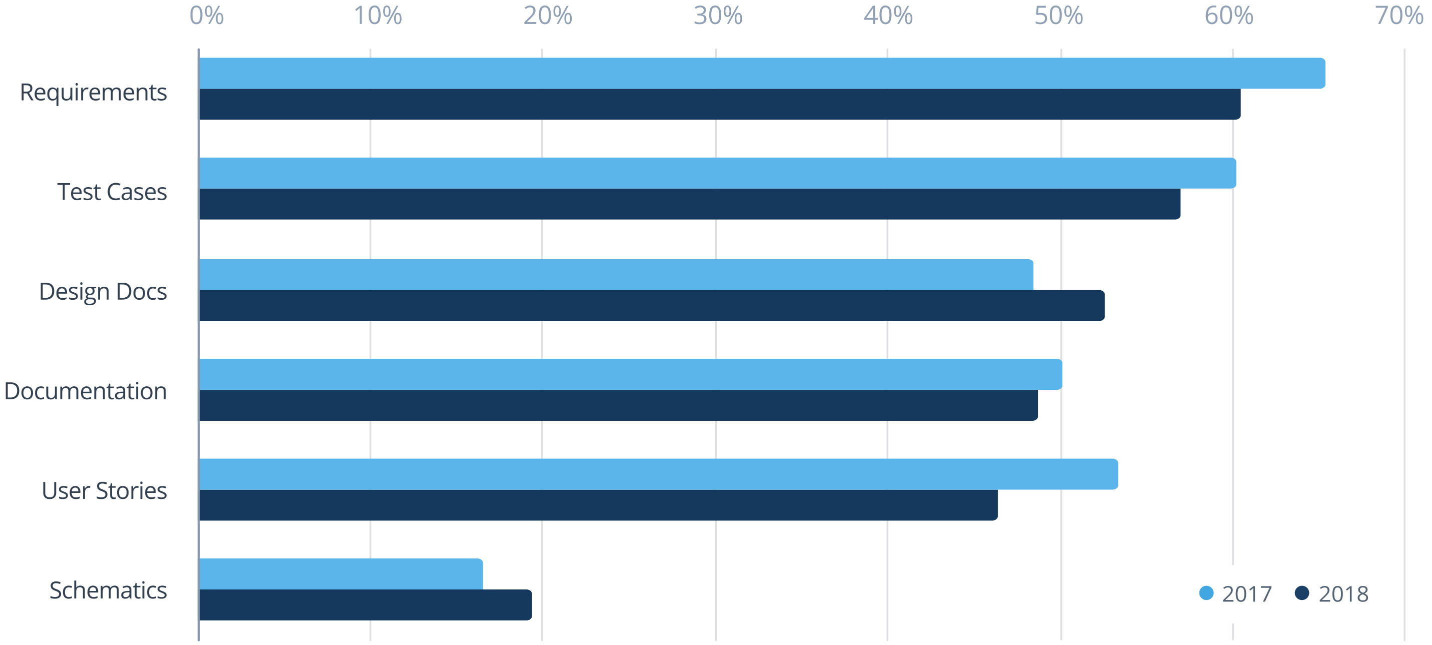
Many people recognize that knowledge transfer across the team is a major benefit to peer review; however, we don't necessarily put programs in place to support and encourage it. We just assume it will happen naturally.

It does, but think how much more so if we were as intentional as the gentleman that spent 9 months doing it. He committed to raising up talent on his team.

Sharing Knowledge to Increase Your Bus Number

It is also important to remember that knowledge-transfer can occur when we review other artifacts as well. In this chart below, we can see that most teams are reviewing requirements, test cases, design documents and more. Document reviews are a key way to build cross-functional consensus on the big picture for a project.

Which of the following artifacts do you review, if any?



n = 561 n = 1102

Are you familiar with the concept of the Bus Number?

Your Bus Number is how many people on your team have to get hit by a bus before no one knows what is going on.

A more positive version of this might be someone winning the lottery and quitting. Regardless, you should always be thinking about training and coaching your replacement. Our goal should be to make it easy for someone to step in and takeover.

So, ask yourself: Who could step in if I were gone?

Take the time to coach them. Hopefully, it is only happening because you've decided to change jobs. The likelihood of being eaten by a shark is 1 in 3,748,067. Being struck by lightning is 1 in 79,745. Falling to your death is 1 in 4,238.[XII] What about the likelihood of leaving a job? 1 in 2 are considering it!

Marcel Schwantes wrote an article for Inc.com that delved into why employees really quit their jobs. He referenced a number of different polls and studies that contain some scary numbers: [XIII]

- 51% of workers are looking to leave their jobs
- 74% of workers are satisfied where they are but 66% of them are still open to new opportunities
- 34% are planning on leaving in the next 12 months
- 44% of Millennials will leave in the next 2 years if they are given the chance

That means it is entirely likely that you will find yourself in the position of trying to piece together a project or supporting something not written by you, or that you'll be the one leaving others to figure out what you have stored in your brain.

It is very important that you take the time to set up a culture of collaboration that sets the expectation of knowledge transfer. Believe it or not, you'll actually retain more employees. People will begin to feel valued and start to take ownership – and that is the next part of the collaboration hierarchy.

Collective Ownership

Ahead 2/3rds

In his book 'Turn the Ship Around', Captain Marquet asked, "What happens when the leader is wrong in a top-down culture? Everyone goes off a cliff." [XIV]

Captain Marquet was promoted, given his own command, and was to take charge of the USS Olympia, a fast attack submarine. Two weeks before taking command, he received a phone call and was informed that he would be taking command of the USS Santa FE, a newer submarine.

The only problem was that Captain Marquet spent a year studying the Olympia. He knew it inside and out, and now he didn't have time to learn everything about the USS Santa Fe. On top of that, his crew was ranked last in all of the measurements the Navy had.

Captain Marquet knew how to bark orders and be in charge of a 2 billion dollar, football-field-sized vessel. After all, he was the Leader.

One of the first orders of business was to shut down the nuclear reactor and have the crew run through a drill. Battery backups came on-line and the sub kept moving forward. To really put the pressure on this slipshod crew, Captain Marquet thought it would be a good idea to push them harder. He decided that running the sub at 2/3rds the max power would force the crew to work harder and faster.

He told the Officer on Deck, "Ahead two-thirds", and the Officer on Deck repeated to the helmsman 'Ahead two-thirds'.

Nothing happened.

The submarine didn't increase speed. When Captain Marquet looked up, he could see the helmsman squirming in his seat.

"Helmsman, what's the problem?", asked Captain Marquet.

"Sir, there is no two-thirds setting." replied the helmsman.

As Marquet describes, organizations tend to have a leader-follower culture instead of a leader-leader culture. We don't typically speak up until it is too late, and we don't encourage people to take ownership and respectively disagree.

In the 2017 Human Capital Trends report by Deloitte Insights, only 14% of executives believe that the traditional organizational model—with hierarchical job levels based on expertise in a specific area—makes their organization highly effective.[II] Instead, leading companies are pushing toward a more flexible, team-centric model.

In the story, Captain Marquet recalls asking the officer on deck if he knew that there was no 2/3rd setting. The Officer replied that he did and Captain Marquet asked,

"Then why did you issue the order?"

The response was, "Because you told me to."

That reminds me of the question we have all been asked by our mothers: "If your friends jumped off a bridge, would you do it to?"

If we don't set clear expectations and encourage individual leadership and accountability, we cannot expect people to take ownership.

James Betteley rephrased Collective ownership in his article "On Collective Ownership and Responsibilities", saying, "I much prefer the concept of shared ownership and collective accountability."[XV]. He tries to encourage all of the members of his team to think the following:

- How does your code build?
- How do the tests execute?
- How good are the tests?
- How good is the code?
- How easy is it to deploy?
- How easy is it to maintain?
- How easy is it to monitor?

As already mentioned, transitioning a team to scrum illustrated the challenge of ownership and setting clear expectations. Part of that ownership and accountability is our ability to measure success.

In the case of development, we might have burndown charts for sprints and release date estimation reports. We do daily standups and hold sprint retrospectives so that we know what people are working on and if there are any blockers. Also, we want to know what's working and what's not.

Without having metrics, we don't really know where we are relative to our goals. One way to capture metrics on your process is to utilize productivity and communication tools that automatically capture that data. Once you have measurements, then your entire team can own their progress.

Communication Channels

The channels that we use to communicate have an immense impact on what is conveyed, how quickly, and with what intent. Communicating in person is the most effective mode, with face-to-face video serving as a suitable substitute for remote teams. In these modes, you have all the available communication expressions, with conversations supplemented by body language.

So much of our work nowadays relies on digital text communication. Before looking at each type, it is important to understand that with text communication comes a negativity bias.

The Negativity Bias

Ellis Friedman, in her article *"3 Ways Your Brain's Negativity Bias Affects How You Communicate"*, discusses how our brains are built to react more strongly to negative perceptions.[XVI]

This is especially true with written communication.
Here are a few examples:

- "That's not what we discussed. Let's talk"
- "Jim, I don't understand this code you've written."
- "When are you going to be home?"

Researchers have studied marriages for years. There has to be a balance of Negative and Positive interactions in any relationship. One of the studies that they did looked at happy couples and how much they fight (negative) vs. having positive interactions.

The research found that there is a ratio that is required between Negative and Positive, 1 to 5, to sustain a relationship long-term.

We are hardwired for negative things to have a significant impact on us. The good news is that we can be aware of it and realize that the majority of the time, people are not out to get us.

Remember what we said about trust. Give trust freely and assume that you are not under attack.

Now as we discuss tools, keep in mind how the negativity bias can affect each one and remember that these tools are all about communication and the foundation should be trust. Keep in mind that the tools that you are using can help or hinder building trust across your team.

Choosing the Right Tools

There are any number of tools that can be used in a collaborative culture. At SmartBear, we use a few extensively. We have a global team so we hold daily standups on Zoom, which enables face-to-face video conferencing. Many of our team members have never actually met face-to-face, but we can still establish relationships and trust by being able to see one another, recognize facial expressions, and hear the tone of voice.

We also use Collaborator for peer reviewing code and documents. Collaborator helps our team build trust because we can see the work that others are doing. It also gives us an opportunity to learn from other people and for other people to learn from us.

For quick messaging, we use Slack and have a channel dedicated to our product. This enables our Support, Sales, Sales Engineers, Development, QA, Product, and Marketing to all communicate openly. It breaks down the functional silos and fosters collaboration.

Choosing the right tools can break down both communication and information silos. One of the biggest trends taking place in manufacturing right now is the idea of the “Digital Thread”. The concept is for there to be a complete record of communication throughout the manufacturing process. This allows the business to share information with the customer, as well as look internally for process improvement areas and establish benchmarks and KPIs.

Take a step back and look at your tools and workflows from a 30,000 foot view. There may be places where you can see significant productivity or cultural gains by adopting the right tool.

The best way to overcome objections to adopting a

new tool is to actively listen to their concerns and use quality questions to uncover the real issue that they are opposed to it. Once you understand their concerns, you will be able to build a case that addresses those concerns. Plus, if you took the time to listen and build trust, they are more likely to trust you.

Fast Feedback

While open communication is critical to a collaborative culture, targeted feedback is also essential. When an artifact or code file is ready to be reviewed, how long does it take for the review to be completed?

Accelerating Peer Reviews

In our 2018 State of Code Review report, we asked how often people participated in different types of reviews. This included Tool-based, Meeting-based and Ad hoc review styles. All of these styles are valuable and can help to shorten the feedback loop.

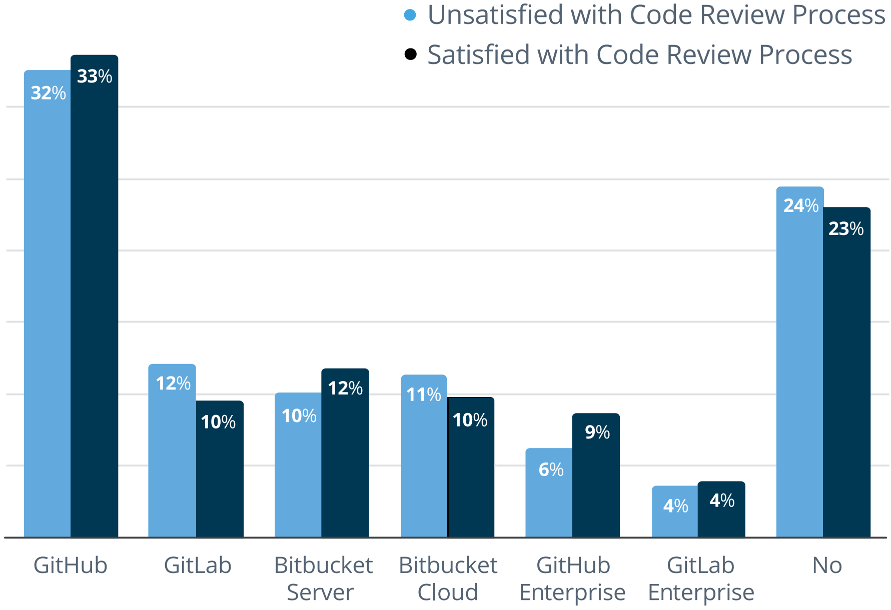
The teams that are conducting code reviews frequently are more likely to meet their release deadlines.

Additionally, we asked 'How often do you participate in a tool-based code review process? Daily and weekly reviews in a tool tended to drive a higher level of satisfaction as it relates to quality.

Tools like GitHub make it easy to request and provide feedback through lightweight peer reviews. 28% of survey respondents said that they were going to be adding a repository management tool in next 12 months and 70% are already using one of these tools.

Simply using one of these tools doesn't mean that you are necessarily going to be more satisfied with your software quality. In fact, we found that use of one of these tools on its own doesn't make a significant difference in software satisfaction.

Are you currently using any of the following repository management tools?



n = 484 **n = 308**

Selecting a dedicated peer review tool that integrates with the pull request workflow allows your team to get the best of both worlds. We use GitHub at SmartBear for repository management, but we use Collaborator for our peer reviews. Collaborator integrates into GitHub, GitLab and Bitbucket to extend the pull request workflow.

Essentially, when a pull request is created, a Collaborator review kicks off automatically. Many organizations need a more defined workflow for their peer reviews and a pull request isn't enough for those organizations. They need to capture metrics for reporting and process improvement as well as have defined roles within the review. Additionally, we tie the GitHub pull request and the Collaborator review to JIRA. This provides us with that Digital Thread so we can reference it any time we need to.

A Collaborative Development Culture

In summary, we've now outlined the Hierarchy of Collaboration for development and QA teams.

Trust: Trust is the foundation. As Roger Staubach said, "If you don't have trust inside your company, then you can't transfer it to your customers."

Clear Expectations: You have to set clear expectations for the individual and the team. Without clear expectations, you can't have working accountability and autonomy.

Collective Ownership: Captain Marquet said, "When the leader is wrong in a top-down culture, everyone goes off a cliff." Help establish collective ownership by creating a leader-leader culture.

Communication Channels: Open up your communication channels. If we are not using our tools effectively, actively listening, and asking quality questions; we will only end up thrashing about and not getting anywhere.

Fast Feedback: Each stage builds to this final level, where your team can focus on accelerating peer reviews and iterating quickly. Faster peer reviews, when paired with all the other collaboration essentials, can shift your development to a new gear.

One Last Note

How many of you know the story about post-it-notes?

Simon Sinek in his book *Leaders Eat Last* tells a story about 3M and how post-it notes came to be. [XVII]

Spencer Silver, a scientist at 3M, was trying to make a strong adhesive, but he failed. Instead, the adhesive was weak. Silver shared his failure with others at 3M in case anyone could think of a use case.

A few years later, Art Fry, another 3M scientist found one. After struggling to keep his bookmark in place at church choir practice, Fry thought of Silver's weak adhesive and the Post-It Note was born.

They didn't work in of a "Me" culture; they worked in a "We" culture.

Foster a collaborative culture that shares work readily. Host internal tech forums where ideas can be shared, including the "screw-ups". We cannot be successful by just writing down and documenting our success and failures. People don't always go looking for the data. We have to share our ideas. We have to make sure that we are providing opportunities for our teams to collaborate.

Collaboration leads to innovation. In fact, one of 3Ms mottos is "Innovation from Interaction".

The result is that 3M holds over 20,000 patents, 80% of which list multiple inventors.

If you think for a minute about the importance of communication, you'll realize that a vast majority of issues arise because of its absence.

Collaborative cultures improve people's lives and allow us to build better quality software.

Regardless of where your team or organization is in the hierarchy, ask yourself this question:

What can I do to build a better culture of collaboration at your organization?

Citations

[I] Covey, Stephen R. The 7 Habits of Highly Effective People: Restoring the Character Ethic. New York: Free Press, 2004. Print.

[II] Global Human Capital Trends (Rep. No. 2017). Deloitte.

[III] Williams, David R. "The Apollo 13 Accident". National Space Science Data Center. NASA. Retrieved December 31, 2012.

[IV] Maslow, A. H. (1943). A theory of human motivation. Psychological Review, 50(4), 370-396.

[V] State of the Global Workplace (2013 ed., Rep.). Gallup.

[VI] Sinek, S. (2013). Start with why: How great leaders inspire everyone to take action. London: Portfolio/Penguin.

[VII] Spainhour, D. (2007). Coach Yourself: A Motivational Guide for Coaches and Leaders. Winston-Salem, NC: EBC Communications.

[VIII] Q&A: Roger Staubach [Interview by J. Kavanagh]. (2005, January 16). Dallas Business Journal.

[IX] Webb, K. E. (2012). The COACH Model for Christian leaders: Powerful leadership skills for solving problems, reaching goals, and developing others. United States: Active Results LLC.

[X] Price, C. In the Beginning. Preaching Today.

[XI] The 2018 State of Code Review (2018), SmartBear.

[XII] “What are the odds of a shark attack?”. The Wildlife Museum.

[XIII] Schwantes, M. (2018, March 7). What Do Employees Want From Their Bosses, Exactly? Inc.

[XIV] Marquet, L. D. (2012). Turn the Ship Around!: How to Create Leadership at Every Level. Greenleaf Book Group.

[XV] Betteley, J. (2014, July 4). On Collective Ownership and Responsibilities. DZone.

[XVI] Friedman, E. (2014, June 12). 3 Ways Your Brain's Negativity Bias Affects How You Communicate. BurrellesLuce.

[XVII] Sinek, S. (2018). Leaders eat last. Place of publication not identified: Portfolio Penguin.

Fostering a Collaborative Culture

Besides checklists and meetings, ensuring that your team is always comfortable and empowered to communicate with each other is critical. Building a successful review process is about more than just adopting a tool, conducting daily reviews, and tracking metrics. It is about fundamentally empowering team members to own their code.

Just being conscious of how your team is communicating, from language to medium, can allow you to anticipate problems before they manifest. A culture of constructive collaboration can be the powerful undercurrent that propels your team's development to the next level, as long as you are willing to make the investment in your process and commit.

SMARTBEAR

The Leader in Software Quality Tools for **Teams**

6.5M+
Users

22K+
Companies

194
Countries

We create the software tools that development, testing, and operations teams use to deliver the highest quality and best performing software possible, shipped at seemingly impossible velocities.

With products for API design and documentation, API and UI level testing, code review, and monitoring across APIs, web, mobile, and desktop applications, we equip every member of your team with tools to ensure quality at every stage of the software cycle.

Collaborator

SMARTBEAR