# Influencing Change


## Authors

Levi Siebens (levi dot siebens at gmail dot com)



## Abstract

Organizational change is a hard thing to accomplish and so few people I've worked with understand how to do that in a sustainable / impactful way. Typically this leads to members of a team focusing on what's "good enough" as opposed to what may be possible. This is due to either a lack of understanding on how to influence or frustration with attempts at influencing that have been ineffective.

This paper walks through the different trials and tribulations I've had in impacting organizational wide change as I've grown throughout my career and how that uniquely impacts the quality assurance discipline. I've had a variety of experiences from small organizations (Mantis, Vertafore) to very large organizations (Amazon, Microsoft). Each had their own unique challenge in impacting change. This paper is not meant to be a "how to become a manager" or "what to do when you have authority" talk. It is meant to share how to create positive change regardless of your role in an organization.


## Biography

*Levi is a Director of Test at Ascentis. Over the past 10 years he has worked at both small (Ascentis / Vertafore / Mantis TGI) and large (Microsoft / Amazon) companies. His experience in testing ranges from video games, mobile, Windows OS and web testing. With his background in Psychology, he not only analyzes systems but also the people that make them up, looking for opportunities to help improve life for the organization he is a part of.*

Design [, testing] and programming are human activities; forget that and all is lost. *--Bjarne Stroustrup [SoftwareQuotes.com]*

*A tester walks up to my desk excitedly and before I can even ask what's going on they begin to tell me about the best automation framework ever. "It will increase our productivity for testing and we can write automation faster than ever!" We discuss the benefits of the framework for a while and once I'm convinced, I leave the tester with this:*

*"Now, go convince the team."*

*The tester looks at me a bit blankly as if they expected me as the lead to go tell the other employees to adopt their ideas. I smile knowing they have a lot to learn about what it really means to create significant change.*

The situation above is fictionalized, but is one I've seen play out again and again. Regardless if it's my idea or it's someone else on my team, the issue that always arises is how does someone take their idea beyond themselves. This may be to a team member, their entire team, their organization or even further. While this is a challenge, anything of significance typically is.

# 1 Why Influence Change?

Testing sits in a unique spot of the development process as it covers validating code, driving features and ensuring the right customer experiences are delivered. In my opinion, without testing it isn't worthwhile to ship a product. Yet this unique responsibility provides a near infinite amount of area to cover. This breadth necessitates finding new and better ways to test, automate and deliver a quality product and the benefits of these improvements are only felt if they are used outside of our own personal testing.

Returning to our opening example, isn't influencing something managers/directors/leads do? The answer of course is, yes. Those who are in a leadership role will influence ideas in a positive direction; however there are many reasons why no one should leave the influencing of ideas up to those in leadership.

## 1.1 It Helps My Career

Regardless of where you are at in your career, the ability to influence change is viewed as a positive differentiator. Want to be promoted in your current role? It doesn't matter if you're at Amazon, Microsoft or somewhere else, it takes the ability to influence change across a team or across an organization. Writing a new framework or creating a new process is fine, but having teams adopt your work is what sets you apart. Driving that kind of change demonstrates that you not only understand the testing discipline, but that you are enough of an expert to create real positive change in your organization.

## 1.2 It Helps My Company

At Amazon, I worked on a test automation framework that had relevance to several other teams. During development, I ran into multiple teams trying to solve the exact same problem at the same time. Like many companies, the teams were treading the same ground in slightly different ways, creating overhead for the individuals on each team. This is an inefficient use of resources, but is also frustrating for individuals once they realize they are doing work that duplicates what someone else's. However, by learning to influence outward, this frustration can be reduced and over time, teams grow closer and learn to distribute the responsibility of the common work across the business.

### 1.3 It Helps The Test Discipline

Many ideas that are good for a team, are good for more than just the team. Whether it's the test discipline at your company or the broader test discipline outside, new ideas are always needed to move the test discipline forward. The way of innovation only comes through small and large ideas driven by individuals who understand how to influence. This opportunity is so rarely taken as many engineers do not understand how to share their ideas broadly in a way that influences those around them.

## 2 Influencing Principles

What are the steps for influencing? There is no definitive list or ordered way to share your ideas with others, however there are a set of principles that can be used to more effectively communicate ideas and ensure they are given the best chance possible.

### 2.1 Understand Your "Customers" and Audience

Good testers are strong at breaking down and understanding complex software systems. Great testers not only understand the system but can put themselves in the mindset of the customer to break the system in the way the customer would break it. This takes a focus on the details of who their customers are and an understanding of all the ways the system will be used.

This attention to detail that is required for understanding customers applies to influencing others as well. The "customer" or audience must be understood at a deep level. This is more than simply finding one thing that they need, it is determining their overarching needs, frustrations and desires. It is only by understanding those things that a person or group can be effectively influenced.

A good example of how not to do this comes from one of my previous SDET (Software Design Engineer in Test / Automation Engineer) roles at a large company. I thought I had a fantastic idea for a remote execution automation framework. This would allow the team to stop running automation on devices under our desks and have a centralized lab in which the tests could be run. I analyzed what I thought was needed without talking to anyone. I even took it one step further and built a prototype (once again without really reaching out to stakeholders) and showed all of this to my boss. It went great! He agreed that this was needed and we should move up our reporting chain as I couldn't build and maintain the infrastructure on my own.

That's when things went wrong.

At the next level, I ran into a manager where from the start I could tell something was off. After a week or so of discussing my proposal, this manager pushed back more and more on my design and architecture. In my engineering mindset, I knew I was right so I argued strong and loud. This only made things worse to where I was finally told to stop work on what I was doing.

By the time these "discussions" were done, I was frustrated as I knew I had a good idea but I could not get past this manager and his desire to use a different framework. So how did I get there? It started with my assumption that my design was the right design for all scenarios. So instead of starting a dialog with the manager to determine what desires he had as a stakeholder, I went to prove why I was right and he was wrong. The more he dug in, the more I dug in, forgetting that I needed his buy-in and influence to move forward.

Most critically, I didn't ask questions during the conversation and didn't get the context before I walked into the meeting with this manager. I didn't stop to try to understand what his concerns may be because I didn't think of him as a "customer". In the long run, my solution was the right one (and was eventually implemented and consumed across 20-30 teams) but I could have presented my information contextually to this person instead of arguing about how I was right. The key here is that those needs and concerns are not what I define, but are what the "customer" thinks and feels.

So what could I have done differently to determine my "customers" needs? I could have followed a similar set of steps that is done in determining how to build a new feature for a product.

1. Identify customers / customer characteristics. Ask questions of those customers, don't start by telling them what they need: "Do you think process X needs to change? Why?" These questions can be leading and can spark conversation, but be careful to not force your own opinion for the solution. The solution should be shaped by this feedback.
2. Identify stakeholders / power brokers and what their desires / opinions are. Know who you would need approval from to move forward (if anyone) and *do not communicate to them till ready*. Premature communication is a great way to get a project killed. Phrase the conversation, "I'm trying to better understand our team's needs on …" or "I've seen a gap in area X, can you share your opinion on that are?" as a way of better understanding their mindset. For help with identifying who may be a stakeholder, see the resources section of this paper.
3. Document this information, it will help you later on when questions arise on why decisions were made a specific way. It allows you to point at tangible feedback from the most important "customers" and have productive conversations around those items.

With this information, it is much easier to move forward and create a plan that not only solves the necessary problems but also addresses the felt needs of your customers.

## 2.2 Have "Vision" for the idea

The concept of a vision is typically something that an executive pushes down to the people doing the real work as a way of exerting influence within an organization. Another version is a vapid statement on a corporate website that is nonsensical. For example, "Our company's vision is to create synergistic opportunities by maximizing the vertical and horizontal integrations across the [InsertIndustryHere] industry". In software development, the concept is used as a patch or a magic fix for getting companies to move in a specific direction. Most of the time, it is a superficial band aid to a complex problem created by those who do not understand what is really going on. Those creating the vision statement feel better by providing "direction", but nothing is truly fixed.

Yet, just because the concept of vision is often used incorrectly, it does not mean that it isn't critical to the success of projects. There is a reason why so often those in leadership attempt to use it as they understand that without vision and direction most projects of significance fail. Granted, a project of one or two people can be accomplished more easily as it is easier to determine the end goal / state. Projects that have a large impact require more people, strong collaboration and a clear vision for what changes could be made.

One of my first times trying to influence, I walked into my managers office to tell him about the wonderful automation framework I had found. My gut told me that this would help us to get past many of the issues we'd had with manually testing the website I was working on. More importantly, the framework I had found was easy to develop with and I felt that would really help the non-technical members on our team. I got really excited about it and went to my manager:

“I’ve found a framework that will let us automate all of our scenarios! Everyone should be using it!”

I’m paraphrasing a bit, but that was my whole pitch. No explanation of why it was better than what we were currently using and no discussion of how it compared to the industry standards. I just had a passion around a framework he had never heard of before and had no vision for how it would change our automation process going forward. Being early in career, my manager was very gracious and worked with me to define out the vision for why it was a significant improvement for us. At this point he encouraged me to demo it to the team. So I did! That’s when the questions started and I really began to understand why having a vision is important. I could list pages of questions but here are the most important ones:

“How does this compare to framework X?”

“What would we do with our existing automation?”

The most pointed question however was, “Why would we use this at all?”

My manager had helped to prep me for some of the questions, but it was then I realized I didn’t have an overarching vision for why our team would want to move to this framework. My teammates were very gracious as they were willing to discuss the merits, even if they didn’t see the need; however not all workplaces are that benevolent. There are many workplaces that would ask the question, “Why would I follow someone and listen to their recommendation when they cannot articulate why the change should be made?”

## 2.3 Start With Answering the “Why?”

At some point, the vision will be ready to present. When presenting to those that will be impacted by the change, it’s important to always start with why the change needs to be made. Upfront communication of this is absolutely critical as this will determine whether your audience (one person or a roomful) will stay engaged. Going back to to the vision example, the audience asked many questions about the why that I hadn’t answered up front. This meant they were spending time breaking down the why for themselves instead of listening to what was being communicated. In addition, I lost some credibility in their minds as I seemed inarticulate on the subject, which simply led to more and more questions.

While finalizing what will be communicated, it is important to think through all of these why questions as much as possible. Walk through the idea with others who understand the area, look for outside perspectives and even pick specific people whom would disagree to get their perspective as well. This helps to refine the idea / vision into something that is focused and addresses the concerns of those listening.

## 2.4 Deal With Questions, By Moving The Conversation Forward

When presenting, it is easy to get caught up in the passion of the idea and become frustrated when questions arise. Many times the idea / vision will seem so straight forward to the presenter, yet endless questions come up. These questions will range from helpful queries to outright hostility. An example of this comes from a time where I introduced a new technology to improve testing across my organization. During the initial demo, one specific person continually asked questions in a more and more aggressive way and I had to think hard about how to respond to each query so I could keep the focus on moving the demo forward. After several rounds of back and forth, I started to respond to a question with a question. For example:

Questioner: “Why should we use this new tool?”

Me: “I’ve laid out some scenarios previously, do you feel like this technology fits those scenarios or are there some concerns you have about it?”

I continued this way for a while, asking questions and metering my responses which provides two things. First, it gave me time to think and reflect on what the other person is asking. It is frustrating to the questioner when their question is misunderstood or they feel they aren't heard as the first response I can think of may not be target their specific concerns. Moving the conversation into a dialog instead of a questions / answer session helps change the tone of the conversation. It also opens up the conversation to others which further anchors the interaction as a sharing of ideas instead of a test of the presenters knowledge.

Second, asking questions will clarify the motives/intentions of the questioner. There are times where the questioner has legitimate concerns which need to be addressed, however there at times (like in the example above) where the questioner is simply looking to undermine others as a form of control. To get at what is really driving the questioner, ask things like, "I feel like you're frustrated about this idea? Can you share a bit what frustrates you about that?" As opposed to, "You're being a jerk, shut up." If the questioner is truly has ill intentions, those will be clear and the speaker can patiently listen without having to argue.

## 2.5 Keep Vision At Center

When a vision is being presented, it is easy to lose focus when there are many questions around either why it should be accomplished or the way in which is should be done. At each question or each proposal of implementation, the vision needs to be kept in mind so that does not become something unintended. There are two questions that are good to ask each step of the way:

1. Is there something that was missed in the original vision which needs to change?
2. Does this question / proposal change the core vision / concept?

If the answer is yes to question one or two, then that is a great time to start a dialog around why the core vision should change as well as what was missed originally. If the answers are no, then it is a chance to take a note about how the question nuances the vision and keep moving forward. Most of the time, a question or implementation proposal tends to fall into question two and can be addressed by the contributors during the implementation stage of the change. It is not something to get caught up in discussing during the initial presentation of the core vision as that can derail the attempt at change. Differentiating the answers to these questions is critical as it will determine how the vision will move forward and in turn the effectiveness at change.

## 2.6 Identify Contributors

Once this vision has been presented, the next step is to identify those who will be more than just a consumer. No large scale project can be completed without collaborating with other individuals. Thus having more individuals to participate is critical to any project. Always look for individuals who are the most interested / passionate about the proposal to identify the best fit for who will help out.

In testing, your customer base is typically the same group as those who would help to drive the project forward. For example, when one individual in the company I was a part of wanted to revise our test planning documentation, the people who were present to push the initiative forward were all consumers of the documentation as well. There may be times that you can find participants from other places as well (dev, PM, etc) however it's always best to start with those who are closest to the tool, process or framework.

The easiest way to identify contributors is by asking the following questions:

1. Who has the most to gain through this change?
2. Who asked productive questions when the vision was presented?
3. Who would like to drive this vision forward?

Each question of these will provide a different group of individuals that will help this move forward. Those who have the most to gain will be the group that can best demonstrate the benefit of the change being enacted. Those who asked productive questions will help refine the vision as it moves along the implementation path. Last, who want to help drive the vision forward are the critical team members necessary to do the work.

## 2.7 Trust Is Everything

I've been in several organizations with varying levels of trust, whether it is an employee's trust in management or management's trust in their employees. The less the trust in the organization, the more limited the ability to influence and change. In other words, without trust people are not willing to change, grow or listen. It cannot be understated that without trust it is impossible to influence change.

Trust is not something that comes for free; it is something that is earned again and again. Each day when someone comes into work, each time they interact with another person, trust is either added to or reduced. If I'm willing to dedicate myself to ensuring the quality of a product is well done, I will increase the trust of those around me. If I am articulate in my understanding of testing best practices and technologies, than others will begin to trust my opinion on where testing should be driven.

However, the reverse is also true. If I am wrong a lot, then my coworkers will begin to question my abilities. If I use others for my own gain as part of my career growth, then they will question my motives. If I always take the easiest tasks and work the least hours, then they will question my dedication and qualifications. Worst of all, once that trust is lost, I can present a vision for what changes must be made, but without trust the changes will be ignored.

## 2.8 10 Years Later…

Fast forward a decade from the automation example I presented earlier, and I find myself as part of a new team that has multiple issues in their quality processes and I am the one in charge of positively influencing the team. The issues on the team range from how testing is accomplished to large gaps in automation coverage and design. I start by asking questions of the team, "What do we need to accomplish first? What's our highest priority? What is our biggest pain point?" The more I dive into these conversations, the better I understand why the team is in the situation they are and I can start to craft solutions that fit the context I find myself in. As an example, the team was working large amounts of overtime due to high dev to test ratio and their approach to dealing with this issue. By understanding the frustration on the team, I was able to influence change in the test processes to ensure quality of the product was maintained without the need for long hours. This led to greater successes for my team (more time to focus on effective testing), for my company (better quality product) and for myself (additional opportunities for growth presented themselves).

# 3 Success! Sustaining Change

The proposal is successful! An entire organization buys into the changes proposed and there are contributors which will help with creating the change. Still, there are a few key things to keep in mind:

1. Drive, drive, drive!

Regardless of who it is, at least one individual needs to take up the mantle of driving and sustaining the change. This means one or more contributors help to hold others accountable for contributing, participating or consuming the changes. Building out a new process is great but if the process is not adopted then nothing has really changed. Thus; whomever is driving the process will work to make visible what participation is required (via email, a roadmap, etc) and ensure that everyone is on the same page about their role, whether they are participants or contributors. Either way the focus is to make sure the change moves forward in a sustainable way.

2. Change happens slowly

Change can take time. The best idea still has to compete with teams realigned with different staff, new company priorities, etc. No team or organization should expect that significant change will happen instantaneously and should set expectations accordingly. These expectations should be understood collectively by the participants and contributors, but also by any “stakeholders” (managers / directors / VPs) to ensure success within the process. These expectations may need to be readjusted as the process progresses to ensure success as the goal is to accomplish the vision in the right way for the organization, not to meet arbitrary deadlines.

3. “They say no plan survives first contact with implementation.” - Andy Weir, The Martian

Whatever plan is made must be flexible and will constantly require a return to the vision of where the plan was meant to go. The goal in change is never to accomplish it in a specific way, instead it’s to move a team / organization in a positive direction. This may mean small changes to the plan or a large scale rework of how the change is to be accomplished. But if these changes move you closer to the vision, then it is worth changing the plan. This is not about catering to different people’s whims, it’s about ensuring a level of flexibility for what really happens when the plan is enacted. Constantly iterate and improve on what the vision is. If it changes as part of the process, then all the better as it will more accurately fit the organization's needs.

# 4 Failure!

Failure to fully accomplish the vision is a very common thing to have happen. There are so many reasons it can happen both for good (right idea / wrong time, no resources, too advanced) and for bad (miscommunication, organization is resistant to change, poor management). The key questions to ask are as follows:

1. Ask, did this fail because the idea / vision needs to change?
2. Who / what is blocking you from moving forward?
    a. Are there ways to solve these problems? If so, try again!

    b. If they cannot be solved today, when can they be solved?
3. Is the idea too large to accomplish all at once? Can this idea be iterated in parts instead of as a whole? (i.e. Can small step improvements be made over time?)
4. What can I learn from this failure?

More often than not, if the vision is continually iterated then a positive change can be made in the organization over time. Most proposals do not simply go from good idea to implemented in the first try. However, if the idea seems to have no support or outright hostility then it is worth questioning whether or not this is the right time for the proposal. There is nothing wrong with learning that a specific vision cannot be accomplished at this point as one can learn a lot about what kind of changes are possible within an organization and what opportunities exist there.

# 5 Conclusion

Regardless of success or failure, it is always worth it to attempt change as more is learned by walking through this process than almost anything else. It takes the engineering skills of a tester as well as the ability to think broadly to propose significant change in an organization. It takes us as individuals stepping out and influencing change to see real improvement in our teams, our organization and our discipline as a whole.

# Bibliography


- Stroustrup, Bjarne. SoftwareQuotes.com, http://www.softwarequotes.com/ShowQuotes.aspx?ID=539&Name=Stroustrup,_Bjarne&Type=Q
- Weir, Andy. 2014. *The Martian*. Broadway Books


# Additional Resources

Below are a list of additional resources that relate to the general topics covered in this paper.


Influencing:
- Cialdini, Robert B. 2006. *Influence: The Psychology of Persuasion*. Harper Business
- Jolles, Rob. 2013: *How To Change Minds: The Art of Influence WIthout Manipulation*. Berrett-Koehler Publishers

Vision:
- Miller, Mark. 2015. *Chess Not Checkers*. Berrett-Koehler

Communication:
- Thompson, George and Jenkins, Jerry. 2013. *Verbal Judo*. William Morrow Paperbacks

Stakeholders:
- Stakeholder Mapping - https://www.smartsheet.com/what-stakeholder-analysis-and-mapping-and-how-do-you-do-it-effectively
- Stakeholder Identification and Analysis - http://archive.hhh.umn.edu/people/jmbryson/pdf/stakeholder_identification_analysis_techniques.pdf