Quality With Hearts Aligned Bolstering Your Quality with Emotional Intelligence

Sophia McKeever

s.mckeever@pokemon.com

Abstract

Quality is more than just numbers in a report, bugs filed, or the tests you run. How do you get teams to align on your qualitative vision? How do you convince others that high quality standards are the right thing to do and how do you compromise to make achieving high quality possible? Emotional intelligence is the key to it all. Quality goes beyond just convincing your team to meet high standards, it's about connecting with them to better understand why they resist and how you can meet them where they're at. Understanding the people you are working with, understanding their frustrations and concerns and looking beyond those surface level discussions to uncover the true reason for push back.

In this, the spiritual successor to Art in Code and Quality, we take a journey deep into that which makes teams build relationships and explore the aspects of emotional intelligence that you can leverage to bolster quality across your organization.

Biography

With over twelve years of industry experience, Sophia McKeever (She/Her) is a self taught Senior Software Development Engineer in Test with an extensive background in test automation and tooling. Her work history includes a small start up, DataSphere Technologies Inc., larger technology companies, both Microsoft Azure and Apple Inc., and, currently, The Pokémon Company International. She is an award winning writer and presenter for her paper, *Art In Code and Quality*, presented at the forty-first annual Pacific Northwest Software Quality Conference in 2023. She has a passion for software quality and strives to bring others along with her as a practitioner of Shine Theory. She holds a certificate in Python Programming from the University of Washington and is well versed in multiple object oriented programming languages.

1 Introduction

When we think of quality, we often picture a numbers game. How many bugs filed, how many escaped, how many incidents following a release, how many tests did we run. It's a chase to ensure that we let as little as possible slip through the cracks while maintaining the highest quality we can. We strive to find that perfect balance between test and release and rarely do we stray from this line of thinking. Yet we tend to overlook an important aspect behind the work that we do. An aspect that transcends numbers and reports and pierces into that which makes us innately human. Emotion.

It goes without saying that behind a development team, there are passionate, hardworking humans who strive are building not just software, applications, and APIs, but also "engage in an act of creation for every line of code they write and task they complete. Creativity, passion, skill, art flowing from finger to key to code file." [1]. They are engaging in creative process, whether consciously or subliminal, to complete whatever is set out before them or whatever they intend to build. It's an expression that is very core to the identity of what makes us human, something that can stir pride within hearts and anger in its critique. Art.

When we frame writing code within the context of making art, it becomes critical to consider the way quality professionals review, analyze, and assess the work of our development partners. We need to not just be able to know and understand the person writing the code, but we also need to be able to understand their emotional state when they wrote it. By applying a modicum of forethought into our approach, we can help ease the burden of delivering critical feedback and difficult messages. However, to achieve this, it is imperative that we engage with empathy and emotional intelligence. Emotional intelligence is a useful tool for being able to not just gauge how someone is feeling and empathize with them but also be able to communicate with them in a clear manner that also considers their needs. This tool is vital to ensure we collaborate effectively and ensure we are heard while also not being seen as rude, condescending, and otherwise negative when doing our work to establish good product quality.

In this paper, we will explore the intricacies of emotional intelligence so we can better understand how it applies to software quality. Using this understanding, it will reexamine the idea that code is art and why being in touch with both our and our partner team member's emotions can help us approach difficult conversations with humility and grace. Finally, it'll explore some real-life scenarios and examine how the application or lack of emotional intelligence affected the situation.

2 Moving Beyond Empathy with Emotional Intelligence

Empathy and Emotional intelligence quite often go together, one cannot have emotional intelligence without empathy, but one can be empathetic without strong emotional intelligence. Not to be confused with each other, Empathy is how we perceive the emotional state of others whilst Emotional Intelligence is our understanding and perception of our own emotions and therefore brings greater understanding to the emotions of others. By utilizing both skills together, we can improve our communication and conflict resolution skills, being able to "foster a deeper level of understanding and trust in our relationships." [2]

Given the artistic aspects of writing code, Emotional Intelligence plays a larger role in understanding how one's feelings affect the code they write and how to analyze the code. The emotions the developer feels get reflected in the code they write, and empathy can help us better identify those emotions. How we respond to those emotions can help us be better partners to the developer and help us better connect with them on a personal level. By engaging Emotional Intelligence, we can identify the feelings our development partners have experienced when they wrote code and better refine our messaging to help ease the delivery of critique, feedback, and accolades to fit their emotional needs.

To begin in this concept, we need to identify what Emotional Intelligence is and understand its role in the empathetic process.

2.1 Defining Emotional Intelligence

Emotional Intelligence can be best described as a knowledge of one's own self and emotions. It is the capability to identify and manage the emotions you experience acknowledging their effect on the relationships you conduct. In her article for Simple Psychology, Mia Belle Frothingham describes it as:

> *"Emotional intelligence refers to the ability to perceive, understand, and manage one's own emotions and relationships. It involves being aware of emotions in oneself and others and using this awareness to guide thinking and behavior. Emotionally intelligent individuals can motivate themselves, read social cues, and build strong relationships." [3]*

In terms of reading social cues, we can learn a lot about an individual's emotional state simply by understanding them and their baseline code writing capability. When a developer writes code in a typical flow state with neutral to positive emotions, it sets that precedent for the standard of code they write. The code is typically focused, concise, and gets the job done. Now picture how the code might change based on their emotion, a developer experiencing happiness or excitement when writing their code can deviate from their baseline but adding more playful variable names, syntactical adjustments, or whimsical comments, in contrast a developer experiencing anger or sadness might write less legible code, or even code that is unfocused. In section three we'll look deeper at these examples, however for the sake of emotional intelligence it is important for us to be able to identify these changes, regardless of how subtle they are, and how they can affect us as the reader.

2.2 Understanding Empathy

It's easy to equate empathy with sympathy or confuse it with emotional intelligence. Empathy can be best described as feeling the same feelings and emotions as another, or simply put, the ability to feel what someone else feels. It can be a means of connecting with others on an emotional level in a way that is relatable without pity or remorse for their situation but rather understanding and supporting them. To quote Olivia Guy-Evans from her Simply Psychology article on empathy, "*Empathy involves sharing another's emotional experience, while sympathy refers to feeling concern for someone in distress without necessarily experiencing their emotions.*" [4] Although empathy and emotional intelligence share a common theme, understanding emotions, empathy is focused on the emotion of others and understanding them.

Engaging empathy in the code review process helps us better understand our partners and their emotional state. It can help us provide thoughtful and critical feedback whilst forming the message in a manner that can better resonate with the developer while in that emotional state. Consider how you might deliver a difficult message to someone who is already feeling hurt. Your approach to delivering that message might be more delicate, using verbiage that would attempt to ease the blow of the bad news. Likewise, for someone who is feeling happy, it might be easier to deliver a difficult message with less nuance while still not disrupting their harmonious state. The same methodologies can be applied to providing critique in code reviews, by paying attention to the code you are reviewing and how the developer is feeling at the time they wrote their code.

3 "My Code Is My Art" - Where Code Meets Emotion

As quality individuals, we excel at critique. We need to approach everything we read and work on with a critical eye to help prevent the escape of bugs to production. This comes with its own unique challenges in the application of empathy and emotional intelligence. It's easy to overlook the idea that writing code is not just a scientific process, but it is also an act of creating art. Think about that engineer who struggles to take critique, who will get upset when critical feedback is given. As frustrating as it can be to work with this individual, by recognizing that pride for what it truly is, art, that developer can become more relatable and help us better connect with them. If we begin to think of software engineers as artists rather than scientists, it makes it easier to empathize with those engineers who take pride in the work they do.

3.1 Writing Code Through the Artistic Lens

As established in *Art In Code and Quality*, the process of writing code can be seen as an act of creation beyond the obvious act of writing it. It was established that the act of writing code can be boiled down into a six-step process.

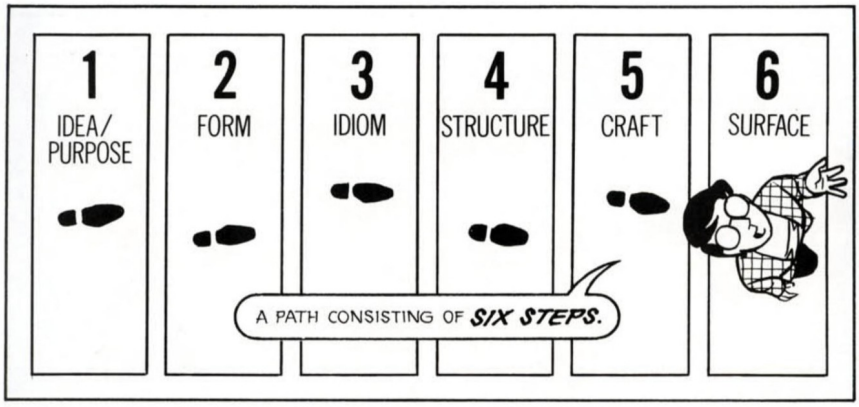

Illustration: Understanding Comics: The Invisible Art by Scott McCloud – Page: 170 – The Six Steps

When we consider writing code using this six step process, it breaks down as follows:

1. Idea/Purpose: The how and why of the work, the project you're working on.
2. Form: How the work will take shape.
3. Idiom: Language and paradigms of the work.
4. Structure: Creating the structure for the project.
5. Craft: Writing the code.
6. Surface: Building and running the code.

In each step of the process, the developer will fold aspects of themselves into the code they write. It is most evident in the craft step, line by line figuring out how to bring the project to life, the developer will flex that creative muscle. In that process, their emotions can reflect in the code they write outside of their typical baseline. For example, when a developer is feeling happy, their code might take a more playful design or might have more creative solutions within it. Alternatively, if they are upset or frustrated the code can be less readable and overly complicated. It can be easy to overlook, however if you pay close attention to the nuances of each pull request, you can notice the structure change given the developer's emotional state.

For example, lets look at two stark examples. In this instance we'll compare two different code snippets. One created for educational purposes and one created when the writer was feeling infuriated. Take note of the nuance of each version of the code. Each screenshot has its own unique properties that define how the emotions affect the code. Both snippets are from the prequel paper, Art in Code and Quality [1].

```python
@patch("example_class.BaseClass._format_error")
@patch("example_class.BaseClass._service_uri")
def test_make_request_error(self, patch_uri, patch_error):
    # Validate we catch when a service returns an unexpected response code.
    # Note the mocking of _service_uri
    patch_uri.return_value = "some uri"
    session = MagicMock()  # Mock session used to ensure requests.session is not called.
    expected = MagicMock()  # Mock used as the result
    expected.status_code = -3
    session.request.return_value = expected

    # Validate Assertion error gets thrown, note no other assertions are made within the `with` statement.
    my_class = BaseClass("prod", "", session)
    with self.assertRaises(AssertionError):
        my_class._make_request("stuff", "in", "a", "box", 200)

    # Validate the expected function calls are made.
    session.request.assert_called_with("stuff", "some uri/in", json="a", timeout="box", verify=True)
    patch_uri.assert_called()
    patch_error.assert_called_with(expected, 200)
```

Figure 1: A Sterile Baseline Created for Educational Purposes.

This particular snippet was created as a sterile baseline. It has no major flaws, it might not be the most efficient code, but it gets the job done. It was created as an intention to help a team learn techniques for building unit tests. The writer made sure to include comments, named variables clearly, and they broke the lines into logical groupings. This is a perfect example of code created with a clear head and neutral emotions.

```python
@patch("datetime.timedelta")
@patch("time.sleep")
def test_wait_some_iterations(self, mock_sleep, mock_delta):
    condition = MagicMock()
    mock_delta.return_value = 1
    se = [False, False, True, True]
    condition.side_effect = se
    scalls = [call(0.5) for _ in range(len(se) - 2)]
    cccalls = [call() for _ in se]
    StaticHelpers.wait_for(condition)
    mock_delta.assert_called_with(seconds=60)
    condition.assert_has_calls(scalls)
    mock_sleep.assert_has_calls(cccalls)
```

Figure 2: A Snippet of Code Written While Infuriated

In this snippet, we see the influence infuriation has while writing code. The writer was attempting to mock an immutable object in Python without realizing that the object itself could not be mocked. As such, they be came more and more angry at the situation, throwing code into the function to try and make it work. We can see their frustration in how frantic the code looks, how nondescript the variable names are, and the fact there is no spacing compared to what was considered their baseline. The code makes little to no sense when attempting to understand it.

3.2 The Human Need for Understanding

Consider the developer who is proud of their work and doesn't take criticism lightly. What drives this individual to be take pride in their work. Could it be because the code is well written? Could it be because they are excited about it? Although these are possible, we cannot overlook the idea that they are proud of what they have created. Part of the reasons good developers a job is because they feel like they either cannot be creative enough [5], or that they feel under appreciated [5]. Given these reasons, there is a stark line connecting code and artistic creation.

When we consider the artistic implications of writing code there is a connection to the artisans emotions and the code they write. Each line influence by the feelings and emotional state they are in as they wrote, often subconsciously, yet present in the code they write. Whether they are aware of the emotions reflecting in their code, there is an opportunity to connect with the individual who wrote the code. When we notice changes in how someone writes their code, there is an opportunity to reach out to them and connect with them. The connection is vital to building upon and improving a relationship with the individual. Building upon that relationship fosters a sense of belonging that can help the individual thrive in their job and feel less like a cog in the machine [6], thus improving overall morale on the team. Teams that are happy with their software quality teams are more likely to work with them and function better as a cohesive unit.

3.3 Transcending Connection

Given that connection can be as important to an individuals physical and mental health just as much as exercise and healthy eating [6], the quality of those connections is important for the enrichment of team interactions. Emotional intelligence is a tool that can help us challenge the expected course of interaction and pierce into the metaphorical heart of the matter at large. When we make an effort to better understand an individual or engage in analysis of the code they have written, we need to be mindful of our own feelings and how they can either affect the feelings of the individual. Being combative or hyper critical when someone is frustrated, angry, or defensive can cause undue distress in the heightened individual and thus close them off to future interactions. However, if we can take a moment to understand why they are so stressed and then take a calm, educative approach in our criticism, we can better deliver a difficult message in a safe and considerate way, minimizing the stress on the reciever.

Additionally, artistic appreciation can help diffuse a situation in which critical feedback is given. When reviewing code, when was the last time you truly complimented a change in your review? Have you ever taken a moment to truly call out something that amused you the code the individual wrote in some way, shape or form? By engaging in a modicum of appreciation, rather than keeping strictly with criticisms and questions in a code review, its easier to connect with the individual and help them feel better understood and appreciated [1]. This act, itself, engages our emotional intelligence as it provides a platform to connect on a deeper, personal level. It proves that you're more engaged then simply trying to point out every single flaw, it shows that you care deeper for the individual than simple perfection.

4 Applied Emotional Intelligence – Theory to Practice

Now that we've established the importance of Emotional Intelligence, lets see how it applies in practice. In this section we encounter two different real world scenarios. The fist scenario is one where Emotional Intelligence is not used and the repercussions of it. The second scenario highlights where it helped unblock progress. Each scenario has the names of the participants anonymized to simple initials to protect their privacy.

4.1 How Ignoring Emotional Intelligence Hinders Progress

Without engaging emotional intelligence we can fall into pitfalls. Lets examine a real life scenario in which Emotional Intelligence would have helped a situation. This scenario involves two individuals, one with the initials L.A. and the other with the initial's S.P. L.A. was a senior QA automation engineer at a trillion dollar tech company tasked with writing test automation frameworks for various applications within the company's mobile ecosystem. She had been putting a lot of effort into building the frameworks in Python and had to begin migrating the frameworks to Swift as part of a technology deprecation. S.P. was brought in by leadership, from a group called the Core Automation team, to help in the transition. L.A. had held pride in the work she was doing, to her it was 'the most inspired work' she had ever done.

When S.P. joined in the effort, he became hyper critical of the work, he belittled and berated code L.A. had written and even more so would use condescending verbiage when commenting on code reviews and documentation L.A. had produced, going so far as to leave comments like "this is a big no-

no, L.A." on things he didn't like. L.A. at first took these comments in stride, trying to find the best in the situation, but they continued to wear on her. At first she tried to reason with S.P., asking him to be more gentle and more respectful in his comments, he refused. All attempts to resolve the conflict were met with deaf ears to the point that she spoke to her manager about the interactions.

Upon hearing her concerns, the manager simply looked to her and told her that it was "a learning opportunity" and to "figure it out". The manager didn't actively listen or engage in emotional intelligence, rather he remained apathetic to the situation. This apathy led to the eventual involvement of the Senior Director of the group. The Senior Director eventually removed S.P. from the project, releasing him back to his former team and reprimanded the manager for not taking action. Unfortunately, the manager didn't take this in stride and took out his frustration on L.A. in a one-on-one, chastising her for going over his head. The final result of this was L.A., once a valuable employee, ended up leaving the company for how she was treated.

Multiple instances of emotional intelligence could have helped resolve this situation. S.P. could have engaged in empathy and tried to be in touch with his own feelings, he could have realized how his attitude was affecting L.A. Had the manager engaged in active listening and empathized with L.A., trying to understand her feelings, he could have seen how bad the situation had gotten and helped mediate. The manager could have also engaged in understanding his own feelings so that in the one-on-one that he could have been better prepared to have a discussion instead of chastising his employee. Emotional intelligence is a powerful tool that could have been engaged at any point to help L.A.'s situation and to keep her as an employee.

4.2 Building Mutual Understanding with Emotional Intelligence

In contrast to the previous section, lets look at a situation where engaging emotional intelligence made a difference. Lets consider a story of two colleagues, O.A. and N.B., both working on a test automation framework based in Python and Selenium. They both had differing opinions on how the framework needed to be built. O.A. wanted to ensure the entire framework would have unit tests covering every component while N.B. thought they were a waste of time.

Over the course of conversations O.A. had tried to make a case for adding the unit tests to every component, explaining that it helps ensure that the framework was making the right calls to given functions and using the right identifiers for locating elements. N.B. felt that the unit tests were excessive and a waste of effort. Their discussions got to an impasse and no progress was being made as both were unwilling to budge on their opinion. Thats when O.A. realized there might be something more than meets the eye. She approached N.B. asking her to lunch and over the time she had a heart to heart with N.B.

During the lunch, N.B. revealed how the team she was working on had been overworked and understaffed. They were struggling to meet tight deadlines and could barely put in the effort to expand tooling as it is. O.A. empathized with N.B., she also realized that her pride was getting in the way of progress. She was able to understand her own emotional state was was causing difficulty for her colleague. Thus, during that lunch, a compromise was struck. Rather than testing all elements and their identifiers, only key elements would be validated ensuring that the framework could remain tested for the most crucial parts while allowing for flexibility in the teams schedule.

Emotional intelligence in this case played a role, helping to unblock progress. It allowed N.B. and O.A. to look beyond their differences and better align on the quality of their project. The realization emotions were getting in the way helped build a compromise between the two and ensured they could continue working together on the project. If O.A. was unable to truly be in touch with her feelings, the project would have likely been a failure. It's important for us to understand our own emotions and how they can affect those around us, in the end it leads to better, more open conversations that can help us come to the right solution.

5 Quality With Hearts Aligned

Quality is not just a numbers game. It depends on all parties understanding how their feelings affect the work that they do and the work of others around them. It also depends on us understanding the feelings of others. Emotional intelligence is the key to help us improve our messaging and comradely within the team. If we are able to understand both our own emotions and the emotions of others we can better deliver difficult messages, reach compromise and ensure our team mates and direct reports feel heard. When we put in the extra effort to understand the emotions of those around us and how they affect the work we do, it allows us to be a better team player and a better deliver difficult messages. It helps us work better, delivering quality, with hearts aligned.

References

1. McKeever, Sophia, 2023. "Art in Code and Quality." *Pacific Northwest Software Quality Conference (PNSQC) 2023 Proceedings* (November): 189-199

2. Rathor, Ruchi, 2023. "The Connection Between Empathy and Emotional Intelligence." Medium – Accessed July 17th, 2025 (https://medium.com/@ruchirathor_23436/the-connection-between-empathy-and-emotional-intelligence-880726c6de6f)

3. Frothingham, Mia Belle, 2024. "Emotional Intelligence (EQ)" Simply Psychology, Accessed July 17th, 2025 (https://www.simplypsychology.org/emotional-intelligence.html)

4. Guy-Evans, Olivia, 2024. "Empathy, Sympathy, And Emotion Regulation: A Meta-Analytic Review." Simply Psychology, Accessed July 17th, 2025 (https://www.simplypsychology.org/empathy-sympathy-and-emotion-regulation-a-meta-analytic-review.html)

5. Forbs Expert Panel. "13 Reasons DEVS Leave Companies (and How to Turn Them Around)." Forbes, September 4, 2020. https://forbes.com/sites/forbestechcouncil/2020/09/04/13-reasons-devs-leave-companies-and-how-to-turn-them-around.

6. "The Importance of Connections on Our Well-Being." Berkeley Exec Ed. Accessed August 06, 2025. https://executive.berkeley.edu/thought-leadership/blog/importance-connections-our-well-being.

7. McCloud, Scott. 1993. *Understanding Comics: The Invisible Art*. New York: Kitchen Sink Press.