

Appendix C - STGT Example

Appendix C: Example of Applying Socio-Technical Grounded Theory (STGT) for Data Analysis

This appendix presents a detailed example of how the Socio-Technical Grounded Theory (STGT) method was applied to analyse raw qualitative data collected from study participants. The table illustrates the step-by-step process of deriving meaningful insights by systematically coding and categorising participant responses.

The columns represent the following:

- **Participant ID:** Identifier for the participant providing the data.
- **Raw Data:** Verbatim excerpts from participant responses.
- **Codes:** Initial labels assigned to segments of raw data that capture key ideas or phenomena.
- **Concepts:** Broader notions formed by grouping related codes.
- **Sub categories:** More abstract groupings that organise concepts into coherent clusters.
- **Categories:** Higher-level groups representing core components of the theory.

This example aims to provide transparency and clarity on the analytical process, demonstrating how raw data was transformed into theoretical constructs that underpin the study's findings. This appendix can be used to better understand the rigorous coding and abstraction procedures central to the STGT approach.

Participant ID	Raw Data	Codes	Concepts	Sub Categories	Categories
P3	Coming to my previous, one of my previous teams, the one where I had to go on a field visit. For that project, I think one thing that really came out was we were able to develop, we were, one thing, we placed a lot of protocols, I'm using that very lightly, but a lot of protocols, and a lot of safeguards around the code about how we process the data and how the validation is done. So that to avoid the situations we were facing.	Improving code quality	Impact on Code Quality	Impact on Product and Quality	Impact on Software Development Prcatises
P22	of course, because at that sprint, like the earlier like how we worked was like, once the developer finish coding, they'll send that over to us. They don't understand what QA does, or they didn't mind of knowing that, but with that scenario that later on, when it comes to finding bugs before sending it to QA, they were able to fix it like they were able to identify it earlier than in the QA process. So that is one thing. one positive point that they were talked about that in the sprint retrospective as well, like we were able to do in this thing, and that's why we get to know, how the QA is thinking and those kind of important factors also came after this situation. So that's that's good thing, I think, from other stakeholders or the team, like scrum master, and you want me to give a point about them as well.				
P22	Yeah, because finding bugs in the early stages of the product development is always like, helpful, and you know it will allow us to have more time to do the chaos testing or documentation or any other work. So I think it's really good. There's a really good impact on product because of that.				
P22	Yes. So the first thing was like, since it was a L1, I mean, high priority bug so, and it was like soon after they released the code to the QA. so I was able to find more scenarios to test, because around that bug, we were able to, earlier we didn't document that scenario that was not in the requirement, but it was crashing the system. So the biggest thing for the as a QA for me, is finding more scenarios to test, yeah, yeah.				
P1	On the other hand, if we are missed up with the requirement and we don't have a clear picture, the quality of the codebase also goes down. because we try to do different things on this, without have a clear understanding. And that also makes the developers' life difficult to maintain the code base and even understanding them. So that is also another consequence.	Reduced quality of the code base			
P1	So because that kind of nature, I mean, simply writing code without knowing the intentions clearly. It always result in bad codebase				
P7	Yeah, eventually, the product also takes a hit, right. Because I still believe that, you know, while back we spoke right. it is, by humans, for humans that were doing any software. let it be AI or whatnot, right. Software is at the end bias for people. And the moment these intricacies come in, there is this sense of belonging that goes off, right. so the team feels that you know what, people here don't understand me, I am just delivering things for the sake of it. And I don't even know who is going to use this. So let me just write a couple of lines of code, just get tasks done and keep moving right. So that at a longer pitcher takes a hit on every possible aspect, right. I mean, the company is at stake, your end users are not happy, the product owners are not happy. And it takes a complete hit because of this sole reason that developers feel this. So they are the ones who get their hands dirty and do the thing, right. You could do a roadmap, you can do things and so on, but software developers are the ones who are actually building the software. And if this lack of empathy comes in, then the root cause of it gets shaken and then the whole point of, you know, software development slowly you know diverges.				
P6	Yes, it did. Like in those modules, yes. People then just wanted to complete the work. They didn't wanted to create test cases and all those, and it suffered in terms of like, the ideal outputs and in such cases.				

## Appendix C - STGT Example

P1	So because that kind of nature, I mean, simply writing code without knowing the intentions clearly. It always result in bad codebase. So that is going further with the team and I mean, whenever we write some code, someone needs to maintain that. So that is also, it may be me or it may be some other developers. And it always make the team's work harder. Because whenever the code base is not that solid, there's much more work to maintain that.	Difficulty of maintaining the code base			
P6	it will definitely suffer. Because, again, if I'm not showing empathy it means like, I care less about the things which is happening and eventually will fall into our bucket, not now, but maybe in future. So, maybe if not, if currently, the team is not suffering, other team will suffer who will maintain this thing or something like that.				
P3	<p>P3</p> <p>And we also built a very robust product that didn't rely on smartphones, etc, which could even rely on say, a dummy phone so. Those were sort of the tangible outcomes, I guess.</p> <p>Speaker 1</p> <p>So basically, it improves the code quality and also the product.</p> <p>P3</p> <p>Exactly Yeah. accessible, more inclusive product, I guess. Yeah.</p>	Developing Better products	Impact on Product Robustness		
P3	Apart from that, I can't think of anything else. Yeah, the product itself, I guess it became more inclusive. That was the big outcome. But apart from that, I can't think of anything else.				
P1	the positive experience Yeah, I if I say there are plenty of positive and negative both ways. the positive side basically comes most of the time as I described earlier, the more we are clear with what they want, and the more they are clear with how the things should be done, and the complexity of the task, then the interaction is always better. And the end output is always better.				
P5	Yeah, I think maybe because, if we are you like in a good mood, like we are not like too much stressed out or thinking about something bad, how they treated us, like that would like you know, sort of like make us stressed out or make us feel down. And sometimes we are not able to focus on you know, whatever the things that we are doing and that could like maybe introduce bugs or you know, what, whatnot. So, and then again, also like, in the middle, we will be like thinking how things happen and all that and that affects our you know, productivity of course.				
P6	it becomes much more easier to actually build a product when when that empathy or when the proper direction is given from top down. So it makes sense like, the product will actually be improved. And the timelines will be met. And everything will move smoothly in terms of completion, and all the plans and tasks which were laid, when we started the project will be completed on time, I think that happens when empathy is shown.				
P7	Okay, okay. So I will, in simple put it as the product quality was good, right. So I'll give a use case of what happened and why this happened. So it was usually with the product owner, that, you know, we get some last minute changes or requirements, and you know, we are bound to change those software requirements, and, you know, adapted quickly. So I had very clearly told that, you know, I mean, this is not a joke, this cannot be done in a couple of hours or days. And this needs a longer effort. So, the product owners and others were empathetic, and they told that, okay, fine, we understand your constraints, take your time, and so on. And even during personal reasons, when software engineering is at the end done by people, right, I mean, we are software developers, we're doing this. So we are not our 100%, every single day, right. So we might have some other, you know, thing going on my end, and so on. I think being empathetic and giving our own space and not micromanaging, understanding a developer's role from our shoes, this has more leverage to practice our craft better right. So giving our own space and time made me do you know, more of optimizations deliver better. And that has helped personally myself, as well as the organisation in the overall product quality. So that's the consequence of being empathetic towards engineers.				
P7	I think it made our job much more at ease, right. So definitely it is not going to directly impact our you know, code or you know, the work that we do. But mentally you are at peace, that, okay, they understand our situation and this is what is our way forward. And let's do the best thing, right, instead of doing something quick and shabby, it's better to wait, clearly communicate. And we also have a backing that people understand us right. so the customer knows what's the situation, the product owner knows that this is what is going on, but they trust us. So that gives us a lot more comfort. And we can focus on our actual work of developing the software. So the whole scrum team felt more, a little more safe than you know, being always in that constant monitoring that, you know, I want to deliver stuff.				
P7	definitely right. So, definitely, there are two impacts that, one is better quality product, right, because you have got the you know space and time, people understand you, there is trust on you. And you also know them better, your customers better that is one reason.				

## Appendix C - STGT Example

P1	On the other hand the negative part, there are some times that we are given some tasks, and we are sometimes not very clear about them, and even we are trying to get more information, we are not getting what I would say more firm and clear information from them. And also the importance of the task, how urgent it is. That kind of thing sometimes, we are not able to get the full picture. In that kind of experience the output is, if it were an urgent task, and we are not giving the enough focus, then the product owner is not happy. And if we do some tasks, and even if we finish it in time, but it's not what the users our stakeholder wants, then they are not happy. So I would say those, those are the two ends of it.	Lack of comprehensive overview			
P1	And if we are thinking if our understanding is quite different compared to what they want, even if we deliver something they are not happy. They tell this, this is not what we expect, we expected and we need these changes to be done on top of that. so we need to do those things in a, in another iteration. And usually in writing software, that is kind of a difficult task. Like we have some picture, and we develop our code base according to that one.	Requiring modifications to developed features			
P9	Just trying to think what happened in that situation. Yeah, well, certain decisions ended up having to be made, and I think what that meant was we just had to come up with another, another solution, or try to, it just takes, it takes longer, I think. because you're trying to get them to understand and meet you in the middle. And you know, if they don't understand the limitation, sometimes it's like, okay, we'll try and come up with something that'll work for you. But oftentimes, you know, that's an iterative process, and then it ends up being that, you know, because they didn't fully understand what it was in the first place. It ends up taking too long, and then they end up again feeling like, feeling not satisfied, I guess, in the product or what was delivered,				
P1	Oh, yeah, I would say in both usability perspective, because if we don't have that understanding, we are not delivering what they want, in terms of and in turns, that is not what the users want. So that part also affected.	Reduced usability			
P5	Yeah, I think maybe because, if we are you like in a good mood, like we are not like too much stressed out or thinking about something bad, how they treated us, like that would like you know, sort of like make us stressed out or make us feel down. And sometimes we are not able to focus on you know, whatever the things that we are doing and that could like maybe introduce bugs or you know, what, whatnot. So, and then again, also like, in the middle, we will be like thinking how things happen and all that and that affects our you know, productivity of course.	Reduced quality of product			
P4	Then not smooth like this way, because they don't understand me and my environment, even my way of working. So they will not get the correct output from me. If it is not that not so. Same as the other way, if I'm not empathetic to others, so I can't express my idea, knowledge, even my knowledge. so not good going forward.				
P2	Oh Yeah. Yeah. Oh, yeah, absolutely. Again, the quality suffers, for sure. Often the delivery schedule suffer. Oh, people will go on leave or work on other things than they should and so on. So yes, it directly affects quality, directly affects quality.				
P2	Oh, yeah, like I said, you know, the product, they get a shoddy product, because people did not put their best into it. So those are yeah, that definitely is, it's the quality that suffers. And there might be delays, delivery delays. And sometimes it's subconscious on the part of the sort of the product groups, in the sense that once they've lost respect of the stakeholder, then they might not unintentionally not do their best.				
P5	Yeah, I think maybe because, if we are you like in a good mood, like we are not like too much stressed out or thinking about something bad, how they treated us, like that would like you know, sort of like make us stressed out or make us feel down. And sometimes we are not able to focus on you know, whatever the things that we are doing and that could like maybe introduce bugs or you know, what, whatnot. So, and then again, also like, in the middle, we will be like thinking how things happen and all that and that affects our you know, productivity of course.				
P2	I would just say that, I would hope that I never, I always delivered a professional, me personally developed, delivered professional work, even if you know, they're not empathetic. But, you know, speaking as a development shop, like I say, that's almost inevitable, is that you will deliver poor quality or poor quality product because not everybody is empathetic.				
P2	Well, like I said, one negative consequence is, to say, I don't want to work with these people again. But, you know, if you're talking about me personally, I at least would like to think that didn't make a difference in the sense that I would always try to produce the best product I can in the circumstances. But I'm sure there's lots of people who will subconsciously perhaps not take that attitude. So again, the negative consequences, one of delivering a poor quality product.				
P5	if you don't have empathy, like negatively to people, we I mean, mentality. And then, of course, that would send you know, without knowingly, like, they might be, you know, coding something, differently. And then, of course, like, you know, productivity could be affected, I guess.				

## Appendix C - STGT Example

P12	In a situation like that, it would be like a usually a rushed, or rushed product or like, something that's like, something that's not technically sound, or doesn't actually solve the problem. Because that can happen. If I'm too prescriptive, I'm saying it, just do it, without regard for how they may want to do it, or their feelings on it, because that can happen. So to get the most out of any one person, especially a developer, it's usually about getting them to understand end goals and like think like the user, when they're doing something, or, or maybe even think about me when I have to demo it to someone. So the so when they're not, when they're not doing that, and if they're just following whatever the ticket says, Yeah, you could totally end up, definitely ended up where it's like, oh, that's, that was kind of short sighted, or they just did the thing that solves the immediate problem, but then we have to, we have to go upgrade that library in three months, because they just pick what was available versus what would have made more sense in a future state. So yeah, I can I can totally have a downy totally have an impact, mostly just around the choices they end up making.				
P13	And also, ultimately the product get affected, like, if I tell something to do afterwards, like, we can do this improve, some, where we give suggestions like improvements, So if the relationship is already strained, that he won't take my suggestions. So ultimately, that affects the product. If it was a good suggestion, so it got wasted. So me not having empathy is one, all that led to this strained relationship, and eventually product got affected.				
P14	Yeah, again, work not being done, that it's aligned with what's needed to be done. delays, extra work. Wrong work being done.				
P13	In the meetings, also, when he was talking, I was very pressured. Like, in the requirement discussion things, If he's talking, like, I was like, Okay, I need to get things in my head. Otherwise, he would, like, be very rude to me if I don't get the work done. So it did affect me, because I didn't understand the requirements correctly. So it led me to have not good mindset, and eventually not good product.				
P7	Yeah I think, I have mostly not being so right. but in very tough deadlines and when there are like, you know, like unrealistic expectations being said that you know, the next Friday is our sprint closure. We'll have to deliver this, complete things and so on. beyond a limit it, so you know that this is not possible, right. So at some point, you tend to resist and you know that this cannot be done, right, I understand your situation, but there is a limit to it. And I will not be able to, you know, empathise with you and do all this, right. So in those times, we take a backseat, but the situation is that they try to then empathise with others or expect empathy from other developers. But that's hardly going to happen, right. Because it's the same case with everyone, everyone has had enough, and they might not actually, and this is completely also a more of a give and take relationship, right. So if the stakeholders are empathetic to you, you also tend to be more empathetic that. Okay, in these situations, they have been more empathetic towards me. So now it's my time to, you know, show it back and you know, give things. But the moment you start feeling that from the other side, you know, there's not enough empathy and support, then you also start believing that, you know, what, this is not my game, and you as a personality, if you're very empathetic, you can go to a limit. But beyond that, you also give up and the whole product starts, you know, breaking where people move out. And, you know, this keeps happening.				
P7	Again, it has broken, right, I mean, since you know, I have not cooperated enough, I have not been understanding them, and, you know, pushing myself further, then the developments are delayed, the customer is unhappy. And, you know, so the other developers also seem to get influenced by this behaviour, that, you know, maybe why am I being too empathetic and doing things when the others in my team are not doing it right. So then it's a ripple effect. So as I told before, it's a give and take between all the stakeholders. if everyone is empathetic, then you feel that, okay, people are all empathetic towards me, I should also be. If not, then it slowly starts breaking and everyone goes their own. So there's this quote on, you know, building bridges and not silos, right. So not everyone wants to do on their own. But ultimately, if people are not empathetic, everyone was very self centric, that I'll do my job and I'll just leave, otherwise, I'll try somewhere else move out and so on. So, this is not a good place to. So it does take an impact on the product at the end.				
P6	it will definitely suffer. Because, again, if I'm not showing empathy it means like, I care less about the things which is happening and eventually will fall into our bucket, not now, but maybe in future. So, maybe if not, if currently, the team is not suffering, other team will suffer who will maintain this thing or something like that.				
P7	Okay, yeah, this quite often happens, right. I mean, software engineering and the scrum model itself, I would still believe that, that is not the best model, proven model for everyone, right. It does suit for people who are intrinsically, you know, like, go getters and who want to do quick things, want to do something and fail fast, and so on. But for not everyone that's a suitable model, right. Someone wants to take their own sweet time and do it the best thing possible. So in a lot of times when people are not empathetic, and they are just not knowing your tech stuff, but want delivery alone, then a lot of times the simplest consequence is burnout, right. So people just don't connect with what I'm doing and why I'm doing it. And then that eventually leads out to maybe stress and just leaving the whole organisation and switching to something else right. this is very commonly happening in a lot of places. Because as you pointed out, the root cause for all this is empathy, right. When the higher ups and the product owners and other stakeholders don't really understand what you do, and if they are not empathetic, they just want, you know, this a trillion dollar business, just keep delivering stuff, and this is what you're supposed to do, then that is where the repercussions of all that is these right. So they don't align with the product, they don't understand the bigger picture, then slowly frustration starts in, and then you know, burnout, and then that eventually leads out to attrition and people leaving the organisation.				

## Appendix C - STGT Example

P7	Yeah, eventually, the product also takes a hit, right. Because I still believe that, you know, while back we spoke right. it is, by humans, for humans that were doing any software. let it be AI or whatnot, right. Software is at the end bias for people. And the moment these intricacies come in, there is this sense of belonging that goes off, right. so the team feels that you know what, people here don't understand me, I am just delivering things for the sake of it. And I don't even know who is going to use this. So let me just write a couple of lines of code, just get tasks done and keep moving right. So that at a longer pitcher takes a hit on every possible aspect, right. I mean, the company is at stake, your end users are not happy, the product owners are not happy. And it takes a complete hit because of this sole reason that developers feel this. So they are the ones who get their hands dirty and do the thing, right. You could do a roadmap, you can do things and so on, but software developers are the ones who are actually building the software. And if this lack of empathy comes in, then the root cause of it gets shaken and then the whole point of, you know, software development slowly you know diverges.				
P6	Yes, it did. Like in those modules, yes. People then just wanted to complete the work. They didn't wanted to create test cases and all those, and it suffered in terms of like, the ideal outputs and in such cases.				
P6	Yes like deliverables and deadlines do suffer from this. If the requirement, if the task is not properly divided, it will actually take longer to complete it. And as I mentioned, like it's not properly showcased to the end user also. So yeah, for sure it suffers a lot.				
P21	Again, I think there might have any issues with the quality of the product due to the lack of testing time.				
P22	Oh, okay, it was very buggy product and customer didn't like it, and they reached out to sending email like, this was really buggy and they didn't like it, so it was not good experience. And for the QA team, it was like we have documented all the bugs and everything. so, but I cannot say we did good, because entire team should be like, should present the quality of the bug, not just developers or not just QA, so we failed it yeah.				
P1	when I'm being able to understand better what they want, and in that kind of scenario, there are some situations that I was able to come up with some suggestions that they also thought better to have in this develop, for example, in this feature. they come up with some requirements, and I try to understand it, and whenever I understand it better, I can come up with some suggestions on top of those things and discuss with them. And they could be even more happy with those	Proactive developer engagement - by offering suggestions to improve product			
P4	P4 Yeah, of course. because I think it makes people talking and discuss. And at that time, we solve problems, and we can talk about different types of solutions. So I think it's really important to have that in, even in tech side environment yeah.  Speaker 1 So it really helps to find the best solution out of many options?  P4 Yes. if we got stuck with the barriers and things in our mind, then we can't express our exact idea inside our mind. So yeah. It's really important to have that.	Helpful in finding the best technical solution			
P11	When they empathised with me, I experienced the ability to, you know, I want to say bargain, that's not really the right word, to negotiate or to discuss with them, the current situation, to propose possible solutions that might require a little bit of their, changing their minds and a little bit of me changing my mind and coming to a consensus. I think when you, you have two empathetic parties, it's much easier to come to a consensus, and it's a resolution.				
P8	Yes. Yes, when one person can understand you better, another person wouldn't hesitate to tell if they have some issue. So then, because rather than keeping something to someone, if you expose it, sometimes you get solutions, and also maybe not a solution to your problem, but indirectly something that could support your problem. So I think yes.				
P9	I guess we could, we just ended up working together to deliver a solution. So I think it was the empathy made it easier for me to also understand what the problem was, because then I was able to fully listen to the BA and then I think that definitely improved the problem solving that we shared and the morale.				
P2	Well, yes, of course, I mean, single out one. I mean, most of the time, I'll generalise here. Most of the time, what you'll end up with, is this desire to do a good job, as good job as possible. You're, like I said, you might throw in extras in there. That you know, we're not, maybe contractually they were kind of discussed, but were not contractually settled. And by the way, you end up making friends that way, too. That's another thing. But you know, I knew you want to be nice to your friends. So you basically do what you do your best for them. So the again, it improves quality, I think, as we mentioned.	Improving product quality			
P2	Well, I'm not sure there's anything new to say on that. It really has to do with, I mean, first of all, there's a quality of what you deliver. But there's also feel like you're willing in those situations to go the extra mile. Meaning that, you know, you might put in a lot more effort into this. So the quality is even better than it will need be, or there's extra functionality in there that you threw in because you were, you know, they were empathetic to you and then you were naturally empathetic to them. That's really all I can come up with at the moment.				
P11	But I think it definitely impacts development and QA. It impacts product as well, right. Because the product team gets what they need or gets what they want. And that allows them to make the customer happy. So it does have, it's just like I said, it has an impact on all areas of the business.				

## Appendix C - STGT Example

P12	And there's a big difference between a developer who just ships the code versus someone who actually like, puts it on their local machine and tries it and then gives feedback to myself or the designers say, Hey, I know we all thought this was an OK thing, but here's what happens, what do you want to do. That person who is empathising with the user in that moment, is maybe they're getting myself and our designer, a little extra work in the moment, but we're not going to remember that extra work in three months. We're just gonna like, Oh, great, We made it a little bit better. So that yeah, that has a big impact, when they're engaged and trying to think of the, the user that we've presented them.				
P12	With that it goes into when a developer is, is thinking about those other perspectives and trying to relate to them, it very often just helps the helps the user more than anyone, because they're trying to put them in the shoes, and maybe they don't understand everything about what that user is going through. that's often someone else's job, it's my job or the designers job. But yeah, it has a big impact when when a developer feels like they have enough context to say, this doesn't solve the problem you think it does. So yeah, other stakeholders not so much, because the, the developer relationship with with product or design, or QA is very direct, right. And because of that, you have a direct like data, very indirect relationship with the user, that they're making the stuff so it's going to impact the user. And empathetic developer or development team will end up because if they make something better for the user, they ultimately make things better by second order effect for marketing or sales or whomever.				
P16	It's a good product. It is, it's a good product. you shouldn't have look at it [Link removed for anonymity]. it's a good product. It's built around people, not technology.				
P13	when they have been empathetic to me, I tried to do my work best. Like, I tried to explore new things, because I'm not afraid. I know, they, they also want me to do good testing and they want, they also want good products. So the ultimately product becomes better. Like, for example, if I don't know something, so for example, if we are discussing on some feature, but I do not know the exact depth of that feature, what I need to, I know up to some extent, but not in the depth. But if developers not empathetic, I'm not going to ask any questions. I'm just trying to test what I know, and what written in the SRS, or the use case, that's all. But if I know my developers are empathetic, and they will explain me, I try to ask questions. So I try to ask questions in different ways and clarify some things that are not written in SRS or the use case, that I think, should be tested. And if they helped me with those, then I try to catch those as well. So that will eventually outcome a good product. Yeah, I think it's the best.				
P13	But as a QA, when we are testing, we get to be in the users' shoes, and think how we are going to use this interface, for example. So if the button is on the top, no it's not too comfortable for the user. If it's in the, at the end of the page, it would be more good because he don't need to scroll it up to submit it. Likewise, we think from the human perspective, end users' perspective so that empathy is a very good when it comes to software developing at the end product, because you won't get user complaints or the stakeholders' complaints from about the software. So having that is very useful I guess.				
P13	I think empathy is basic human quality that everyone should have. like, if we are dealing with humans, we should have empathy. And when it comes to any, any industry or any thing, the empathy builds good relationships. when it comes to software developing, I would say empathy is empathy plays a really huge role. Because in, although we do have developers and QA roles separated now, but all we are doing the same thing, like we are building a product. So in order to be, in order to achieve that, within the right time, with the good quality, and with really user friendly features, we need to have empathy. because otherwise we won't be able to work as a team. And sometimes you make very good product for the user when you have empathy.				
P17	But you know, certainly getting a better quality result, I think was noticed. So the times that they showed empathy towards the work I was doing, or what I was involved in, you know, other than being motivating, it certainly did, I think add to the quality of the work as well.				
P14	Yeah, cause they'll try and design systems that actually work for their end users.				
P3	And deliverables also, like for better quality, because you're not rushing it, you're making sure that everything was up to standard because you're not in a rush. So the deliverables are also of a bit of quality and yeah, build tested and all of that.				
P3	And that will all, that, it will always lead to a better product from our end as well.				
P7	I think that has also put them at, you know, a more trustworthy situation that he is more dependable, and you know, he's able to correlate with my job difficulties as well. So I think overall, apart from the product and quality, these type of empathetic conversations has created a more personal value, right. So the team bonding and you know, within the team that has improved quite a bit. And I think that is the biggest takeaway from this yeah.				
P8	Because of my empathy, again, the same thing, that positive impact, we have been able to do the right thing and deliver on time.				

## Appendix C - STGT Example

P6	So basically, as I mentioned, developers rarely think about it. But there were few occurrences where we need to actually take an account user's perspective into the picture. And particularly there was like a related to human aspects, where we have one of the stakeholders who were actually provided us a story point that, when they were actually showcasing a demo to a customer, they noticed that the customer provided some colour problems to them. And it turns out, like one of the individuals was colorblind there. So based on that, when it came back to us, we tried to actually improve the product, we tried to improve the accessibility features from the UI so that there's there's this sort of problem doesn't occur. So yeah, this was one of the like rare occurrences, where we actually tried to improve the product based on like, what problems the user was facing.				
P6	I think whenever, from the development point of view, if you showed empathy, it had tried to improve the product in itself, and like they were positive results.				
P6	But it is essential in the long run, because then issues can come from customers, and they will be unhappy. And then company will be suffering from the problems. But I think in all cases, it it actually makes a developer kind of a very good in terms of doing his job, and dividing or planning his job in long run.				
P10	So I would say I've saved some business cost, and then the the reputation of the team, you know, it's, it's not a mistake of a developer, but, but I would say these are unexpected issues that happen in production. So we have a different environment. I would say this is kind of an environment issue that needs to be fixed within the team, or it could have impacted the the overall quality of the product could have been impacted. It could have lead to the monetary expenses and the reputation of the team. So, so that's, that's what been saved when I showed empathy towards the situation.				
P20	Well, in the end, when the developers came back, they ended up delivering and the quality was high. So I think that and I think there was some goodwill in the team. So I think it's one of those things where if you're empathetic, it causes you some some difficulty. But later on, people try to make good on that.				
P20	Um, yeah, look, it's been it's been improved, it just also means that we all we work well together, out. So yes, I think it's more about working well together, collaboratively. And in turn, that means you're less stressed. And you can just sort of, you know, get thing, get things done better. Rather than sitting there worrying about being stressed and why everyone's upset. That just puts a bad mood on to everything.				
P21	For that situation, we didn't get any impact for the product, as I completed testing by working in extra hours.				
P22	Yeah, we actually the customers were able to get a quality product which is not creating any crash or anything that's yeah quality improved.				
P19	I can give you another like a story. because we build up, when we build up system, at the beginning, after we finish that, I told everyone I just do a little bit reflection. I tell everyone why we miss, why we made a mistake because during the development I found that our authentic which means when we give the different stakeholders the whole structure of the software, there's something missing in that part. and who find that is our, is [the developer] find that, is not me. Although the prototype is designed by myself and [the developer] told me, I think we missed some things like and I do the double check. And then a tricky part is when I, at the beginning, I thought everyone can understand my my prototype because they got experience. And after this, after my mistake happen, I thought okay, probably I need to list all those documents as as detailed as much detail as possible to give them details. at least if I write it down, they will read it, and they will understand that. So for those software, I think for [the developer], I don't know whether this is his work approach. sometimes you give them the document, they will not read it. They just based on their understanding to design. And then after that mistake happen, and we just have another conversation. Since that time, now I just try my best to write down all those details for for him, and like a, like a point by point in bullet point to list all the details. So which means okay, in a, in a documents add the screenshots for those prototypes, and give them the links for the prototype to explain the like the interaction between those buttons and all those forms, and then also list the details, what I really want in a document. So that's why I think I can use this approach to let him feel better. At least I try my best to explain, explain it. Because usually when we have a meeting with our stakeholder, I thought, because of the time like you're only like a one to two hours meeting, someone will lost in that meeting. In particular, the meeting content is like nothing related to them, especially for those software developers. So they do these things based on requirements. And they show their capability and ability or rather skill. It just everyone everytime I just asking, What do you think, can you, can you do that? And his answer is yes, I can do that. But the final result is sometimes it's totally different without those documents with all those prototypes, but at least I find, okay, I can use the conversation or communicate with them to show my empathy to talk about how I solved the problem based on their personnel. And also I can list on details in a document to let them know or help them as a kind of instruction.				
P21	I think empathetically highlighted the previous issue during planning meetings and retro meetings. Understanding the impact I had on project timelines and quality, and it helps me to handle my workload, as well. So I think it impacts to the delivery quality as well. Yeah.				



## Appendix C - STGT Example

P21	<p>Yes, I think as management decided to have another QA member, I think product quality also we can focus on product quality more than before.</p> <p>Speaker 1 Do you think product quality has improved because of that?</p> <p>P21 Yes. Yeah.</p>				
P3	No I guess it sort of, for a long time, the product we were working on wasn't very inclusive, it was a bit limited to a certain customer base who had access to an advanced set of technologies. So yeah, that was the big disadvantage that we faced.	Developing products with limited inclusiveness			
P2	So they're not empathetic, the point is, I think you don't expect them to be empathetic. Again, we're talking, when we talk about say stakeholders, and we're talking more about the the very high level where there's the consumer and the producer, right. At that, at that level, you know, it's different if I'm talking to say the testing people, a stakeholders right. You know there, there's first of all, we're all in the same company and so on, so that makes a difference too. And you're more of much ready to empathise with them. But if, if you don't have empathise, say with the testers, that doesn't change things. Might you still have to deliver something that the testers will pass, whereas if you're talking to a customer and you deliver something that they say doesn't meet the contractual obligations, you still have to do it and but you do you do a minimum right. So I feel like the desirability of the product is diminished. Maybe if not the quality but the desirability of the product might be diminished. These are subtle differences if you like.	Developing less desirable products			
P14	I think the same, where they have a broader worldview view of the end customer.	Improved understanding of end users			
P12	Because if the developer is empathetic towards the goals or the situation, like they just understand it better, and they, I find they ask better questions. And that's really what gets you to it, right. Like if they, they're engaged, and they understand, and then they ask questions you just improve, right, iteratively.				
P12	In that case, that's very much a situation where if I'm doing my job right, they're empathetic towards the user, so yeah, anytime where the developer has a good understanding of what the user is trying to accomplish, I found those projects work so much better, because they're trying to, they're trying to empathise, they put themselves in the shoes of the person who's actually going through and clicking.				
P12	With that it goes into when a developer is, is thinking about those other perspectives and trying to relate to them, it very often just helps the helps the user more than anyone, because they're trying to put them in the shoes, and maybe they don't understand everything about what that user is going through. that's often someone else's job, it's my job or the designers job. But yeah, it has a big impact when when a developer feels like they have enough context to say, this doesn't solve the problem you think it does. So yeah, other stakeholders not so much, because the, the developer relationship with with product or design, or QA is very direct, right. And because of that, you have a direct like data, very indirect relationship with the user, that they're making the stuff so it's going to impact the user. And empathetic developer or development team will end up because if they make something better for the user, they ultimately make things better by second order effect for marketing or sales or whomever.				
P18	Yeah, look, I think. I think, yeah, I have. Look, I find that they're actually probably 2 things. So not necessarily direct interactions with, with myself. But when developers are interacting with other people in the business, I think there are probably two aspects to it. I think, definitely, when they're empathising, with the end user, I think we more often than not get better outcomes, because they are actually thinking, you know, what, what is this end user going through? What problem am I actually solving for this end user, and they can come up with, you know, more innovative ideas, or they can come up with, you know, I guess, you know, better solutions to the problem that the end user is having. Yeah, so for the most part.				
P12	I think it leads into them being more empathetic towards the goals or the user or me. So it kind of goes into the same cycle of the the other question, right. Where it's like, the developer, especially in most, like Agile Scrum teams, right, the developer, it's kind of closed off from the business side of it, or the or the user side. So if I'm empathetic towards them, and representing the user in that moment, and then they become empathetic towards the user, or even the business. Yeah, you get the same like, positive cycle going as when they're empathetic, because they think, I find it triggers, I find it triggers their empathy.	Increased empathy towards user			
P12	So the so when they're not, when they're not doing that, and if they're just following whatever the ticket says, Yeah, you could totally end up, definitely ended up where it's like, oh, that's, that was kind of short sighted, or they just did the thing that solves the immediate problem, but then we have to, we have to go upgrade that library in three months, because they just pick what was available versus what would have made more sense in a future state. So yeah, I can I can totally have a downy totally have an impact, mostly just around the choices they end up making.	Suboptimal solutions and rework			



## Appendix C - STGT Example

P12	in this in this context, being empathetic towards me could mean being empathetic to like a user I'm representing. But yeah, because sometimes like the designer or the UX person, or myself, maybe even like the developers' manager, or their scrum master, he just made like not a great decision about a solution where the requirements were not great, or whatever the case is, right, Like, it's, yeah, we can do options A, B, or C, and we chose C, because it's fine. The developer isn't thinking about the user, they're just following the requirement, then maybe see what not a great idea, after all. Maybe they may they, if they're if they're only thinking about finishing the job, then, you know, the job will be finished until they finish the spec, and then you'll realise Oh, it's not, it's not actually great, like in practice. So you have to do rework. And that takes more time.				
P6	I don't think like, if we, as I mentioned, like, if we didn't think about these empathy, then like, customer came back saying, Okay, you didn't take this into account, you should have taken this into account. And then we need to actually work on it. So it's like, if we didn't show empathy, then it will be a problem for us. Like we need to actually work on tasks extra in terms of actually like adding the user's requirement into the product.				
P15	yeah, as I mentioned before, actually does not really affect the developer a lot, for the product development progress, because the thing is that it might have an impact in later. Well, they want to scale up. because sometimes they want to finish that thing we don't really take into consider of other platform. (It may fix that.) But without a very close connection with different product also from, it might later really hard to scale, and maybe integrating with others, other application is very hard to do that. So it might be seems fine at this stage. Because you know, sometimes they might work, just kind of very flexible, because when we finish one thing in one hour, you kind of finish a lot if it's urgent, in one hour as well. So I will say, it won't affect a lot at this stage on a project project as well, but it might have a impact for later development. Actually, it's a lot. Sometimes you have to redevelop the whole software, if you want to scale up or in a different way.				
P12	But a developer who is thinking about the user or thinking about the end goal of the requirements, will say like, oh, this, I don't think this is doing what the product owner as a designer, the manager thought it was going to do, why don't I try that again? And then oh, you know, this is not doing and then that really helps a project.	Improved solution quality			
P12	With that it goes into when a developer is, is thinking about those other perspectives and trying to relate to them, it very often just helps the helps the user more than anyone, because they're trying to put them in the shoes, and maybe they don't understand everything about what that user is going through. that's often someone else's job, it's my job or the designers job. But yeah, it has a big impact when when a developer feels like they have enough context to say, this doesn't solve the problem you think it does. So yeah, other stakeholders not so much, because the, the developer relationship with with product or design, or QA is very direct, right. And because of that, you have a direct like data, very indirect relationship with the user, that they're making the stuff so it's going to impact the user. And empathetic developer or development team will end up because if they make something better for the user, they ultimately make things better by second order effect for marketing or sales or whomever.				
P18	Yeah, look, I think. I think, yeah, I have. Look, I find that they're actually probably 2 things. So not necessarily direct interactions with, with myself. But when developers are interacting with other people in the business, I think there are probably two aspects to it. I think, definitely, when they're empathising, with the end user, I think we more often than not get better outcomes, because they are actually thinking, you know, what, what is this end user going through? What problem am I actually solving for this end user, and they can come up with, you know, more innovative ideas, or they can come up with, you know, I guess, you know, better solutions to the problem that the end user is having. Yeah, so for the most part.				
P19	because when we have a better communication with each other, we decrease the cost to be honest, and a shorten time and find a better solution for the workflow. So because at the beginning, we just, we got our own workflow, our way to work, but during this, I will say this empathy part and empathy meeting, everyone happy to share their skills, their mindset, to the whole project. so everyone feel better and to get to learn from each other, at least, we know each other is how they're all their ideas, how their way to think of. So I think this is very good way to improve our efficiency, and come up, come up with ideas. So the result is better. So I just do a little bit calculation, I remember that, before I join the team, they will literally cost like at least three months to six months to finish your project. currently we will finish a project, a similar project, use the similar module, which only use including test, we only need to like use one and a half a month. So we shorten the time, and we decreased the cost. (And for less costly part because of the developer), because they gave us very good I think solution for the sustainability. So I think these guys know how to do that. But without good communication with them, and without a trust, I think, yeah, in this conversation empathy, all around the trust, I want to, I would, I would say that. because of the conversation or this communication, we build up trust with, because we care about each other because of empathy. So we build up the trust. So with this trust, we shorten the time, we decreased the cost. And we build really good relationship, the whole feeling about the team is better.				
P22	Yeah, because finding bugs in the early stages of the product development is always like, helpful, and you know it will allow us to have more time to do the chaos testing or documentation or any other work. So I think it's really good. There's a really good impact on product because of that.				

## Appendix C - STGT Example

P18	So like, clicks in our software is gonna be really big, right. If we come up with a user pattern, that's 10 clicks, and the developer just does it, and they ship it great. You hit the spec, but our customer may not be happy with the end result. When a developer is thinking, it's like, oh, well, even just as a simple thing, oh, you know what, We could save like three three clicks by doing it this other way. And oh, wow, I want that save the time for the user, yes. Well, let's get that that makes a big difference. And then if you just do that again and again and again, because you're always thinking about the user.	Enhanced user experience			
P11	So sometimes when they lack empathy, you walk out of a meeting, and you're deadlocked, and you cannot get the information you need, you cannot come to the, come to a common solution to a problem, what generally happens, which is unfortunate.	Difficulty of finding a common solution			
P12	And the like, on the negative side of it, you know, very much just like an interpersonal thing. where it's like, oh well, that didn't feel so great. Like I don't, you know, some days when you have a bunch of those, or it's like, Oh, you don't get what I'm trying to do. And that can just feel very draining, and then kind of drains creativity.	Stifled creativity			
P18	I think Yeah. Okay. It's, it's a typical thing, like, I find, you know, the more separated that developers come from the end users and from the stakeholders and whatnot. You see it a lot, when you know, you deal with outsourcing in, multiple levels of outsourcing as well, where, you know, the ultimate person developing something may have zero contact, even with the company that they're actually developing something for I think you inevitably end up with a slower pace of change and a product that's not necessarily what the business actually needed, or wanted in the first place. Because that empathy wasn't there to actually understand what the end user would go through and what the end user actually needed to achieve. Because the developer didn't have any contact with with that end customer and end user. So I think there are definitely negative outcomes associated with not empathising with the with your end user.	Products that do not meet user needs			
P18	However, there are there's the potential that the reverse can happen. you know, that when, you know, you might be, you know, like I was saying earlier on, you know, there's often a separation between developers and end users now. There could have been a BA sitting in the middle of Solution Architect, different layers of people sitting in between the business end or the end user and the developer. And I think, you know, what, what can occasionally happen, that doesn't happen often, but occasionally happen is, you know, the BA may have got the requirements wrong, or the design may have been wrong that went to the developer. The developer started building things and you know, there may be progressed, you know, 75% through the build. And then they might actually start questioning why it's been built like this. They might reach out to the end user and say, why is this happening. They'll have a discussion and then developer uh this is all garbage and throw out what had been built and start from scratch. And now sometimes, obviously, that may be a better outcome, because potentially the product will be better at the end of the day. But it will potentially take longer and cost more to actually develop. So, you know, there can be a negative impact on the cost and the speed to market, those cases.				
P14	yeah it didn't align with the core needs. It might deliver the feature but doesn't actually meet the underlying purpose of the feature.				
P13	Yes, because I have seen some. because when we are empathetic, specially QA, When we are empathetic, we try to think of user perspective right. for example if we give a mock screen, that we need to develop this. so developer will do that. Even he thinks, if even he, if he has empathy, he will design according to some usability things.	Increased usability due to developer empathy towards user			
P9	Yeah, okay, I admit sometimes I don't understand designers. Most of the time, I do show empathy towards designers because, also partly because I feel like they're underappreciated generally by the team, and also because I am interested in what they're interested in, like I said, usability. But I think sometimes when I get a review and it's very nitpicky, it then becomes really hard to understand why. Asking me to make certain changes or whatever, like, if they're like, oh, this needs to be two pixels to the left, I'm like, why? And then I feel like there's a bit of a disconnect sometimes with like, what's important in the product yeah.	Disconnect in product priorities			
P9	I definitely think empathy is required to be good at your job, and in terms of, Like, performance wise, if you are empathetic, you're more likely to understand the needs of the customer. I feel like it's not just enough for an employee to understand like the organization's like mission statement or values. If you're not empathetic towards the other stakeholders, then you know this, you won't come up with the best solution to a problem. So I guess that's what I think on performance and what was the other one,	Better understanding of user needs			
P19	Yeah, miscommunication with end user and convert the wrong requirements, and also cost money.	Incorrect requirements			