

Project: Investigating User Feedback from a Crowd in Requirements Management in Software Ecosystems

Report created by Mobile46 on 23/06/2024

Document Report – Grouped by: Document Groups

All (20) documents

Groupless

20 Documents:

1 P1 INTERVIEW TRANSCRIPT

Text document, 11 quotations

P1 INTERVIEW TRANSCRIPT Researcher: OK, now we start recording. In this first part of the interview, I will introduce some concepts to align our understanding of them, OK? Interviewee: OK. Researcher: The first concept I wanted to talk to you about is software ecosystems. According to the Lungu concept, we consider a software ecosystem as a group of projects that co-evolve in the same environment. In general, this environment is formed by a common technological platform, where several software projects have been developed, with some interconnection through integrations and so on. From this, we have a group of actors involved in this ecosystem—for example, the central organization that controls this platform. The central organization has a team of developers who develop products and the platform. Furthermore, there are also external developers, who are those who either create a project or some integration, and there are end users who are those who use the products and the platform in general, which is also used by external developers. The example I have given is Amazon, right? Which is the closed software ecosystem, which has several products developed there. There's Prime, which is streaming, and AWS and others, where they co-evolve in the same environment. OK? Interviewee: Yes. Researcher: The second concept I will bring you is requirements management, a requirements engineering process focusing on monitoring the requirements life cycle. Requirements management focuses mainly on the changes in these requirements during their life cycle. So, some activities related to requirements management will be asked about in a moment, such as communication of requirements, change control, negotiation of these requirements, and prioritization. I imagine this is a concept you know well, right? Interviewee: Yes, definitely. Researcher: The third concept is crowd-based requirements engineering or CrowdRE. CrowdRE is an umbrella term for utilizing automated or semi-automated approaches to collecting and analyzing feedback from users who are part of a crowd. A crowd is a large group of current or potential software product users. A crowd in software ecosystems is a sizeable

heterogeneous user base that provides requirements across multiple channels with diverse stakeholders. An example is the context of mobile app development, where you can use the app store and other communication channels to rate apps with stars and write feedback about them. From these comments, you can get a lot of interesting information and even identify requirements and changes to requirements. However, evaluating thousands of comments manually is a difficult task. These are the three concepts that we will discuss in this interview. Are they clear to you? Did you already know all three? Interviewee: Yes, I already know these concepts in practice, but I didn't know the name Crowd-RE. Researcher: The next part of the interview is for me to ask you some questions and hear more about your experience managing requirements in software ecosystems using feedback from many users. As you told me before, your company is big, right? You have partners, customers, and users, which generates a crowd in your ecosystems. Interviewee: Yes, I worked with many people, and many people use our systems. Researcher: So, the first question is, do you usually consider this crowd when identifying requirements or changes to requirements? If so, what approach, be it technique, method, or tool, do you use to identify these requirements or changes when considering the crowd? Interviewee: First, one of our systems has a built-in feedback system. You can set the number of stars at the end of use and describe some improvements. We have already researched this same system. Our UX team puts some users to use and analyzes their behavior. We also do A/B testing with different features to identify requirements. We make some features available to a specific group of customers and others available to another group. In other words, a certain group of customers has a greater need for some features, and others have a greater need for others. So, we develop these features and release them to them. So, we develop these features and release them to them. After it matures, right? Then we release it to everyone and give everyone the option to use it, understand? So, there are several possibilities here, right? We are also using gamification to reach this crowd and identify requirements. We reward users who give feedback with a bonus. Researcher: For example, how do you analyze the feedback for this feedback channel that you mentioned? How do you identify requirements or changes? You must receive a lot of comments, right? Interviewee: We do it by grouping. We group the features by similarity. Then, we see and analyze where it came from. We try to simulate all these features and understand whether they bring value to the client's business or not. And then we classify. Researcher: This grouping is done manually. You receive it. You see a related subject. Interviewee: Yes, the UX team. We schedule a meeting every 15 days to carry out this type of analysis. However, there is another issue, sometimes we have five levels of clients within the same client, right? There is the plaintiff, the sponsor. So, we also see the weights, right? What do these people do, right? So, there is also this type of analysis, right? This varies from client to client. There is ABC customer analysis, etc. Researcher: It's cool that you talked about this now because we can now move on to the next question, which is... Do you then consider this crowd to prioritize these requirements? You haven't mentioned much yet about weights for different types of clients, correct? How does it happen to you? In addition to these weights, do you use any prioritization approach? Interviewee: It depends on the project. We use the weighted shortest job first, a SAFe algorithm. We also created another alternating algorithm that uses weight and value in prioritization. We can have several criteria.

We explain these criteria. The criteria can be time, working hours, effort, more value for the end user, more value for the internal user, cost, and benefit. So, we show all the possible criteria and try to put together an arrangement of these criteria. We also prioritize directly with the customer often. There is no single approach. There are numerical approaches that we use. We also use a statistical approach. It depends a lot on the situation. I'm not talking about a product. I'm talking about a series of products. Yes, both those who are selling in the market and developing or developing innovation. There are products that I will tell you are not on the market. There are MBAs involved in projects with the government. I'll give you an example. The [omitted] and the [omitted] asked for an agenda, right? To go with my CEO. So there is also a product aimed at them, you know? So, prioritization will appear, for example, of [omitted], of [omitted]. Another product from the legal area is different. So, I have a first degree, right? Prioritizing the product, I have a second degree and am a judge who prioritizes the product. So, it will depend a lot on the layers, so the product has a single core, but I start changing their interfaces, right? And his behaviors, depending on the situation. That kind of thing will vary a lot. I have another product that the end user determines, which is much cooler because the person will use it daily. Researcher: There are many interested parties involved, right? Interviewee: Yes, yes, you have a vast number of operators. So users can automatically give feedback on a product of this type. I will give you an example. People talk a lot about chatbots. We do not use a chatbot. We use an artificial intelligence tool that does what the chatbot does but is much more intelligent. We can identify this entire level of feedback in an omnichannel service through several channels. Use your own artificial intelligence for training. It manages to capture this well. So, the concepts we have for certain things, the little boxes break. They're not little boxes anymore, right? The various little boxes are no longer several little boxes. Today we start to unite them into a bigger box, that's what we're doing. Researcher: The next question is related to requirements negotiation. Is crowd considered in requirement negotiation? Thinking that requirement negotiation occurs mainly when there are conflicting requirements. These conflicts can exist between these end users, of which there are many, but also between these customers, which in your case are also many due to the multiple systems. The question, then, is how this negotiation occurs. Do you have any approach to this and consider yourselves to be that crowd? Interviewee: Often, the client has no experience. This type of prioritization is not even used to create innovative products. What do we do? We first have to conceptualize it and present the various possible criteria. We can have several criteria. We explain these criteria. The criteria can be time, working hours, effort, more value for the end user, more value for the internal user, cost, and benefit. So we show all the possible criteria and try to create an arrangement with them. An arrangement of these criteria. I'll give you a very simple example. Often, in the legal field, a judge has to have career achievements. He has to have several achievements to be able to progress in his career. A project manager also has to have deliverables. So, middle and senior management want deliveries to have merits, and underneath, this is not reflected and only increases the work and bureaucracy. So, this also happens, we have to map this environment, the influence of this environment to see what weights we will give on performance. Another thing we also talk about is our service capacity. You have to put the majority of representatives or the players at the

table or their representatives. And try to reach a consensus on a roadmap to deliver these requirements according to an order. Requirements conflicts are normal. We are already used to dealing with conflicts and offer criteria for their resolution. We offer win-win approaches, considering weight and hierarchy. And then we have to align this. And, of course, we try not just to comply. We have to show our work. We go to this meeting with the work already done, with the requirements, to show the gains and pros and cons of the gains. Sometimes, it's not the best gain. But sometimes, the strategy with the customer must also be very close. Soon, he will say that something cannot be done now, despite giving a lot of value, and that they are not prepared for it here. This happens directly; sometimes, the client prioritizes something they are not prepared for, and we have to align with them weekly to change the priority or even the scope of delivery and the roadmap. This happens straight away. Researcher: And what ends up influencing the other client, as you said in this group, which are related, right? Interviewee: This always has some impact, either positive or negative, there may be idleness on your team or they fail to deliver another client or serve another client within the deadline. This can also impact if you are using shared teams. Not even in dedicated teams. Then we have problems with team idleness, so you have to work on that. You have to look for other things with value to deliver and avoid idleness, right? In this sense, some are well organized, and others are not, right? We already leave a pre-backlog there, right? At that point, that's it, and people just start carrying on the activities. Then you do a pulled kanban, right? Let's say it like that, right? Then, we become much more aware in this sense to act. It depends on maturity because they are different companies, products, and areas, right? And despite everything, many different people working. Both in terms of technical and soft skill maturity levels, correct? It's very variable, right? Researcher: So, from what I understand, as you are very much at this intersection for each meeting, you define a point where they want to negotiate, how they want to prioritize, with what form of identification there, which is most appropriate and from That's what leaves people, right? As you said, you become unnecessary because they have already understood what they have absorbed and will execute it alone if that is what you follow. By far, correct? Interviewee: Exactly. I help them create a work process, define their methods, and define the criteria. I follow them very closely at the beginning, leave them to walk alone, and always make a point of continuous improvement. In other words, revisit this periodically, at least once a month. The minimum suggested, right? To see if the way we are working continues to make sense or if we must change. I also strongly encourage this; continuous improvement is the most important thing at all our meetings. The question is whether the way you work makes sense. Researcher: Ok. The next one is related to actual user feedback. How does the communication of requirements occur in the ecosystem you are engaged in? At some point in the interview, you commented on this. So, this question has already been answered a little, you said that you have a direct feedback channel in one of the products and I imagine you must have it in others too. You talked about omnichannel service and the use of artificial intelligence that goes beyond a chatbot. You also talked about meetings with the team and players to make decisions and everything else, they basically work like this, are there other channels that you use? Open channels like forums? How does requirements communication work considering the ecosystem crowd? Interviewee: We also use some [omitted] tools, for example,

[omitted]. So there are some forums [omitted] that we can do. Sometimes, we manage to prepare a questionnaire and launch it internally or to the client. The client can also apply this questionnaire internally in their company. As the whole group is a consultancy in itself and has some products, we are used to this ease of communication with the client. So, we try to encourage the client to do internal research, or we do the research ourselves. So, we ask if they want to do the user research internally or prefer that we do it. We can also open externally because some group initiatives are really very open. We have forums where users suggest some improvements. These are things that are open to the general public. We participate in some working groups to develop some standards. It is pretty varied, including labor legislation. A company in the group works only with HR, which is very involved. So, she participates in forums, including other legal and technical issues. Researcher: Ok. The next question concerns automated or semi-automated tools to deal with requirements in your operating context. Do you know or have you used any automated or semi-automated tools to deal with user feedback from a crowd in the context of the ecosystems you are engaged in? For example, some tools mine comments made about applications available on Android, of which there are thousands. From the results of these tools, from the analysis of the results, it is possible to identify trends, showing the evolution trend of an application, what people like, what needs to be changed, what is most commented on. Therefore, do you know or use these tools in your activities? Interviewee: Yes, we do. We are talking about the following: today, we talk about big data analytics. But we are already moving to a third level, which is machine learning, data driving, driven by data, but now it is machine learning itself that makes decisions. We have clients who are already much more advanced. Many decisions are made by artificial intelligence itself, which already recognizes, sees, and makes decisions. Other clients still prefer to leave the decision and requirements very much in their hands. We have two scenarios: one where the AI already makes the decisions, and it already infers to us that what is there is terrible. Some projects have already been set up this way, and others have not. In others, they suggest and give us suggestions, and then people will decide. It will be more data driving, machine learning, and decision-making. We have both automated and semi-automated, and the other that has nothing, has people looking at graphs and talking, it has everything. Researcher: And in this automated context, do you have an example? Interviewee: We have a client in the United States, for example, who has some campaigns, and as his campaigns are launched, he invests in advertising. So, he invests in online advertising, which costs a lot, and he measures the level of conversions and there are purchases. From there, he invests less or more, depending on the channel he enters. So, it also maps advertisements across various channels, and it maps which channels are being most effective, and then, it redirects investment or intensifies investment in that channel, the system does this on its own and it has several smaller and larger parameters to work with. Like the financial scholarship, but he does it online. He analyzes and changes how advertising is invested in that channel. So this is very automated. It's an American client. Researcher: And how does semi-automated work? Interviewee: Yes, there are some scenarios that we will discuss with the client. One of the projects that I am directly involved in today has this scenario. However, there is another one that doesn't. In the other case, the client is only informed, receives the information, and configures what

he wants; he makes the decision, and we are not involved. We only developed the solution that sent him the opportunities and suggestions, and he made the final decision—who acts on the system? Researcher: OK, let's move on to the last question now, which is related to these three ways that you do it: automated, semi-automated, and manual. From what you said, there are projects that you consider the crowd and others that you don't. Any project must have a smaller representative set of stakeholders, correct? Interviewee: Yes Researcher: Right! The final question is whether you believe that considering the crowd benefits you or better the activities you carry out in requirements management in this scenario, or whether it gets in the way. What is your opinion on considering the crowd for requirements management activities in software ecosystems? Interviewee: The crowd is important, but the final decision does not belong to the crowd. We use the crowd to feed a system that provides indicators. The crowd is part of the decision making, but the final decision never belongs to the crowd. Researcher: Another question I have about automated approaches? Those that consider the pre-determined parameters for decision-making. How do they work? Interviewee: It depends on the client. Some consider it. It has learning feedback parameters and learning reinforcement, which is what we call it. He accepts feedback and learns more about it, increasing his learning about the whole, and others don't; the client doesn't insist on giving more feedback; it's completely automated. Researcher: Got it. OK, [omitted], we've reached the end. Thank you for your time. Thank you very much. I'm going to finish the recording, and I would like to know if you could recommend someone who could participate in this interview as well, someone who has experience in this subject.

11 Quotations:

1:2 ¶ 25 in P1 INTERVIEW TRANSCRIPT

one of our systems has a built-in feedback system. You can set the number of stars at the end of use and describe some improvements

3 Codes:

● _Mechanisms / ● Built-in user feedback modules / ● Requirements change identification

1:3 ¶ 25 in P1 INTERVIEW TRANSCRIPT

Our UX team puts some users to use and analyzes their behavior. We also do A/B testing with different features to identify requirements. We make some features available to a specific group of customers and others available to another group. In other words, a certain group of customers has a greater need for some features, and others have a greater need for others. So, we develop these features and release them to them

3 Codes:

● _Approaches / ● A/B testing / ● Requirements change identification

1:4 ¶ 25 in P1 INTERVIEW TRANSCRIPT

We are also using gamification to reach this crowd and identify requirements. We reward users who give feedback with a bonus.

3 Codes:

- _Mechanisms / ● Gamification / ● Requirements change identification

1:5 ¶ 38 in P1 INTERVIEW TRANSCRIPT

We use the weighted shortest job first, a SAFe algorithm. We also created another alternating algorithm that uses weight and value in prioritization. We can have several criteria. We explain these criteria. The criteria can be time, working hours, effort, more value for the end user, more value for the internal user, cost, and benefit. So, we show all the possible criteria and try to put together an arrangement of these criteria.

3 Codes:

- _Approaches / ● Multi-criteria decision-making methods / ● Requirements prioritization

1:6 ¶ 42 in P1 INTERVIEW TRANSCRIPT

users can automatically give feedback on a product of this type. I will give you an example. People talk a lot about chatbots. We do not use a chatbot. We use an artificial intelligence tool that does what the chatbot does but is much more intelligent. We can identify this entire level of feedback in an omnichannel service through several channels.

3 Codes:

- _Approaches / ● Artificial intelligence techniques / ● Requirements change identification

1:7 ¶ 46 in P1 INTERVIEW TRANSCRIPT

We can have several criteria. We explain these criteria. The criteria can be time, working hours, effort, more value for the end user, more value for the internal user, cost, and benefit. So we show all the possible criteria and try to create an arrangement with them. An arrangement of these criteria

3 Codes:

- _Approaches / ● Multi-criteria decision-making methods / ● Requirements negotiation

1:8 ¶ 46 in P1 INTERVIEW TRANSCRIPT

Requirements conflicts are normal. We are already used to dealing with conflicts and offer criteria for their resolution. We offer win-win approaches, considering weight and hierarchy.

3 Codes:

- _Approaches / ● Requirements negotiation / ● WinWin

1:9 ¶ 58 in P1 INTERVIEW TRANSCRIPT

We also use some [omitted] tools, for example, [omitted]. So there are some forums [omitted] that we can do.

3 Codes:

- _Mechanisms / ● Forums / ● Requirements change identification

1:10 ¶ 59 in P1 INTERVIEW TRANSCRIPT

we manage to prepare a questionnaire and launch it internally or to the client. The client can also apply this questionnaire internally in their company.

3 Codes:

- _Mechanisms / ● Questionnaires / ● Requirements change identification

1:11 ¶ 59 in P1 INTERVIEW TRANSCRIPT

We have forums where users suggest some improvements. These are things that are open to the general public.

3 Codes:

- _Mechanisms / ● Forums / ● Requirements change identification

1:12 ¶ 79 in P1 INTERVIEW TRANSCRIPT

The crowd is important, but the final decision does not belong to the crowd. We use the crowd to feed a system that provides indicators. The crowd is part of the decision making, but the final decision never belongs to the crowd.

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

2 P2 INTERVIEW TRANSCRIPT

Text document, 6 quotations

P2 INTERVIEW TRANSCRIPT Researcher: Okay, the recording started. Well, let's begin part one of the interview, where I present some important concepts for the rest of the interview. I start with a definition, perhaps the simplest, of software ecosystems, which can be considered a group of software projects developed and co-evolved in the same environment. So, when we think about a software ecosystem, the idea is to have a common technological platform and a group of projects that co-evolve around it. In general, some actors participate in this ecosystem, such as the central organization responsible for this platform, internal developers who are part of this central organization, and some external developers who can also develop projects around this platform. And the end users who benefit from all the products generated in this environment. An example of a software ecosystem we have is Amazon, which,

in general, most people are familiar with. Amazon had an e-commerce, which still exists, right? This e-commerce gained popularity, and today, it also has several other products, right? Prime is streaming, and AWS is the cloud service that co-evolves in the same Amazon environment in the same ecosystem. I am also giving as an example the [omitted] who are part of the [omitted] cooperation. I don't know if your institute is part of it, too. Interviewee: We are part of [omitted]'s cooperation, which is similar to that of [omitted]. Researcher: Got it. So, it's another cooperation, but it must be organized similarly to an ecosystem. Right? Interviewee: Yes, but when you say that projects are developed and co-evolved in the same environment, development can be external, right? Researcher: Exactly that. Interviewee: They will co-evolve in an integrated way, but development can be external, right? I can even have a system in my ecosystem that has independent developers. Researcher: To a certain extent, yes. They may be independent but need a common technological platform for several reasons, including competitive advantage. The example I talked about [omitted] works like this. There is the central organization, which is [omitted], and other institutions with autonomy. Interviewee: We have the same example in [omitted]. The unified system of public administration. It is the system that manages the entire administrative part of [omitted], and now the academic part is also being implemented in [omitted], and it is a system developed in cooperation with [omitted]. They evolve, and we evolve, so there is this cooperation, and other institutes are entering. I think it's similar to what you said there, right? Researcher: It's very similar, and I was interested in learning more. This [omitted] ecosystem has more than 90 institutions; if I'm not mistaken, there are [omitted] also involved. Interviewee: It's because [omitted], an initiative began for institutes to integrate to develop systems some time ago. And then several [omitted] institutes integrated to develop [omitted] and to develop [omitted]. So I think it's in the same line, yes. Researcher: Okay, now I will move on to requirements management. In general, requirements management is the process that accompanies planning and development related to requirement changes that occur in the life cycle of a project. So, in general, it will be related to monitoring changes, traceability, and version control of requirements. On the next slide are some activities directly related to requirements management. That's a little bit of what I'm also interested in in this research: How these requirements are communicated, how changes are controlled, and how these requirements are negotiated and prioritized. So, in the questions section, we'll talk about these activities in a little while, okay? Interviewee: Yes. Researcher: The third concept is crowd-based requirements engineering. So, when I was talking about online user feedback at the beginning, there is now an area that has been discussed since 2015. Crowd-based requirements engineering uses automated or semi-automated approaches to collect and analyze much information. This large amount of information is called user feedback in the literature, which helps in requirement engineering activities. Overall, the practical example I'm showing in the interviews is that of a smartphone's app stores, where you can start an app and comment about it, whether the app has any bugs or issues, and where it could improve. For an application with a large number of users, like Facebook, this will generate many reviews and comments, making it difficult for you to analyze this amount of feedback manually. Crowd-based requirements engineering, among other scenarios, acts this way. OK? Interviewee: OK. Researcher: After explaining the concepts and

presenting the examples, I would like to confirm with you whether you consider that you are already or have been engaged in a software ecosystem and whether you have already received or analyzed feedback from a crowd. Interviewee: Yes, I've already participated in an ecosystem and received feedback from a crowd. Researcher: On the form, you indicated that you did not receive feedback from a crowd. Interviewee: I only considered my current context. However, as you explain now, I realized that in [omitted]'s projects, there was a crowd involved. Researcher: Got it. So, now we move on to the second part, in the second part I will ask you some questions about your experience in crowd-based requirements engineering in software ecosystems, according to the concepts I presented to you previously. This crowd, as you mentioned just now, as you gave or gave an example, can be related to employees of an institution, in a university, there you have hundreds of employees who use a system and will be affected by a system that will be developed. However, some stakeholders are representatives, managers, some representation of students, and technicians from whom we elicit the requirements. Interviewee: Ok Researcher: The first question is, have you ever considered a large and heterogeneous set of users, what we call a crowd, to identify requirements or changes to requirements in the context of your software ecosystem? Interviewee: Yes, in the projects I had with [omitted]. We developed software in partnership with all [omitted]. So we had all the institutes as users and [omitted]. Researcher: What kind of approach did you use to identify requirements from this crowd, these other institutes, and partnerships? How did that work? Interviewee: We received feedback from users on our forums. We had an environment to organize. We had a kind of library of requirements to store and prioritize. We used [omitted], an open-source task management system, but we also adapted it to link requirements to the task and have a way to prioritize, schedule, and evaluate. We created a requirement library in Redmine. So, yes, I have already worked with a large group of users that I can consider the multitude, considering several institutes from [omitted] and [omitted] when we started to develop [omitted] and built a library of requirements in a tool for inserting requirements, evaluating and prioritizing, in addition to having online meetings. Researcher: You used this forum you mentioned just now, right? Interviewee: And the forums, too, but we needed this library as there were many users. Researcher: I will focus a little on the forum because I was curious. About the forum, I participated a lot in generating messages. This analysis of the forums was done manually, meaning someone would have to go there and read the comments. How did it work? Interviewee: There were a lot of posts on the forums, and the analysis was manual, yes, but some users preferred to convey their needs in a meeting; they didn't have that incentive to use the forum. So, I feel like forums don't replace the need for another strategy for requirements elicitation, you know? Be it group meetings or brainstorming meetings, even if you do them with user subgroups. I think the forum is critical, but it will not replace group meetings. I think they have to use both strategies. Researcher: Using these two strategies, could you identify changes to the forum? Someone could be saying something interesting there on the forum and in the meetings. From there, you fed this tool, right? Is that what you mentioned about prioritizing, controlling, and making the connection between the requirement and task? Interviewee: That's right, but I feel that when we have a large number of users, what you call a crowd, we end up needing to have some critical stakeholders because we know that prioritizing

requirements and negotiating requirements are highly complex tasks and each stakeholder will have their particular interests. So, we end up needing to have some key stakeholders to put the hammer down. And we had bigger managers from the institutes and [omitted], so more focused meetings, with fewer stakeholders to, in the end, prioritize and negotiate. Researcher: Interesting. You ended up getting a little ahead of my two questions about prioritization and requirements negotiation. In other words, they are already in tune with what you just said. I will leave the two questions on the slide so we can comment on them, as they are very related. Taking into account that you have already considered a crowd to identify requirements. The next question is, have you considered this same crowd or another crowd to prioritize requirements? You didn't talk much about the issue of prioritizing requirements, taking into account many people, which is difficult and complex. So, have you considered prioritizing this large number of people? And if so, did you use any approach for this? Interviewee: So, to prioritize, we met with a small set of users, with key users. Then, it couldn't be with everyone; everyone could insert requirements in the forum and suggest requirements in meetings that we held with subgroups. But at the time, this was all in the tool, in our library. We had to hold a meeting with a group of strategic users. With institute managers and [omitted] managers, then, smaller meetings, not with crowds. To prioritize, to negotiate, and to prioritize. Researcher: Right, but thinking a little about the structure of [omitted], of [omitted], even managers are a small crowd, right? Dozens of managers of [omitted] can also generate a certain complexity in prioritization, right? So, how did this prioritization happen with managers? Interviewee: So, we had a small group. A smaller group that were not the managers of all the institutes were managers of groups of institutes by area. Did you understand? He was a manager by area, not by institutes. The institutes elected representatives of the regions. Researcher: So it was more by consensus that the requirements were prioritized in the meeting? Interviewee: We met with the small group and then, in the end, [omitted] also gave the hammer. There had to be a manager there [omitted] to lead. Let's say we had to have technical approaches and method tools to prioritize. So, our approach was to reduce this crowd significantly, reach a minimal number of key users and key stakeholders, and have strategic representatives who, in this case, would be from [omitted] who would give a final word; they would always be Minerva's vote. In negotiation and prioritization, we did not use anything other than group meetings with a small group and strategic users with Minerva's vote. Researcher: I understand and taking into consideration that negotiating with a lot of people is not an easy task and neither is prioritizing with a lot of people, as you said in an ecosystem context, everyone will have their own objective, right? So, this makes it difficult and much more complex. Some initiatives, such as those in open-source communities, use voting. They open it up to the community to vote. They are now moving on to the next question about requirements negotiation. Some people have treated prioritization and negotiation the same way. Interviewee: They are very interconnected. Negotiation and prioritization Researcher: I usually differentiate them by giving an example. When we are negotiating requirements, we generally have conflicting requirements. I have a requirement that defines the color blue for something and another requirement that defines the color red. You can't handle both. It's not a question of deciding which one I will paint: blue first and then red. If I paint it blue, I

will please a stakeholder, which in a software ecosystem could be a company, and if I paint it red, I will please another stakeholder and displease the first. If I displease the first one, he could leave the ecosystem or stop contributing as he used to, which could affect the health of the ecosystem. So, when discussing requirements negotiation, we focus more on these conflicts. Do you understand? Including conflicts related to requirements, changes are the focus of requirements management. Did these conflicts exist in the context of your ecosystem? How did you deal with them? Interviewee: From my experiences, as far as I remember, we have dealt with these possible conflicts within the group of strategic stakeholders. We validated these requirements, which users requested on the forum in this group. What we thought made sense we defined. Researcher: Got it. Interviewee: That's why negotiations in this sense you're talking about are more difficult for me to give my opinion on. In the ecosystem, we had this group that validated the requirements of this crowd. Researcher: Okay. Interviewee: I don't remember many conflicting requirements. I remember requirements that the group of strategic stakeholders could deem unnecessary could be invalidated. Researcher: Got it. These are the last questions. They are related to interaction with the crowd. How did you get user feedback from these people on these projects from a crowd? And how did you communicate the feedback you received? In other words, possible requirements or changes to requirements. You gave an example a while ago, saying you received it on the forum. So, I consider that this question has already started to be answered, but were there other communication channels you used to receive this feedback besides the forums and meetings you mentioned? And how did it work internally between these partners, these groups, and group representatives? Interviewee: We received feedback on the forum and in-person and online meetings. This was before the pandemic. This was my experience. So we traveled a lot to the institutes and to [omitted], to [omitted]. Researcher: Got it. Interviewee: We received this feedback through forums and face-to-face and online meetings. How did the communication of requirements go, considering this crowd? Researcher: You had previously mentioned a little in the context of software ecosystems and partnerships. With [omitted] and the other partners in the development of the projects, where there was not so much conflict and which had a heterogeneous crowd of users. How did you communicate requirements to your team? For the [omitted]? And for the others involved? Interviewee: So, we communicated through this environment that I told you about, the environment that we adapted for task-oriented requirements management. We transformed this environment into a requirements library that linked the task to the requirement. The planning of the development team's tasks was based on the requirements. The development team was geographically distributed, interacting through the platform. Researcher: Right. This information about the platform and the team being distributed is really cool, which is also a very peculiar characteristic of this context of the development of software ecosystems. From this point, the penultimate question is whether you know or have used any automated or semi-automated approach to managing requirements in this ecosystem context. Interviewee: So, we used the adapted Redmine. [omitted] developed a library to manage requirements and artifacts, which was [omitted] in his master's degree. At the time, we were also trying to acquire a tool to manage requirements, but we ended up using Redmine itself, associating tasks with requirements to manage these requirements. I think it was the

most automated I've ever gotten. Researcher: Right! Today, there is a discussion in the context of crowds and automated approaches that involve mining requirements repositories, which are requirements libraries, as you called them. From this mining and the use of more automated approaches, use this feedback available in different repositories, whether on Twitter or in another tool such as GitHub itself, through the issues that are opened or in the forum itself, such as you set the example here. In other words, the community goes to the app stores to comment. From that, we can use these automated approaches to analyze the text, sometimes sentiment analysis, for example, to identify trends in what that community is asking for, the most reported bugs, or even what has a greater or lesser impact. Interviewee: We don't use it, but it's very interesting. Regarding what you said about the traceability area, we achieved good traceability of the requirements for project planning, project management, and task management in this project. Researcher: Making this task x requirement relationship? Interviewee: Yes, yes. Researcher: The last question is whether you consider that feedback from a large and heterogeneous set of people, that is, a crowd, benefits requirements management in the context of a software ecosystem. If yes or if no, why? Interviewee: Does crowd feedback benefit requirements management in software ecosystems? If you have a control group I believe so, if you don't have this group I think you can't manage it. Researcher: Got it. Interviewee: User feedback from a crowd benefits requirements management only if there is a small strategic control group of stakeholders to manage this feedback. Otherwise, I think it doesn't benefit because it becomes disorganized. Researcher: Got it. You recently mentioned that you used a forum but didn't consider that using the forum alone would help without a more restricted group of stakeholders. You said that one complements the other and that both can be used in an ecosystem context. That's what you said. This can benefit if there is some control mechanism for this amount of information, correct? Interviewee: As long as there is a control mechanism, a strategic control group. Yes. Researcher: Right. We have reached the end of the interview. I want to thank you for your time for this interview. Your answers certainly helped me a lot. I don't know if, now that you've gone through the interview, you would have someone to recommend, someone who worked with you or who could answer these questions. Interviewee: Someone from the requirements and ecosystems area that you could interview. A [omitted]. Didn't [omitted] speak to [omitted]? Researcher: He nominated her. I think she might be on vacation. I emailed her last week, but she hasn't responded. Interviewee: Tell me again that she takes a long time to read. There's [omitted] too, the one who did her master's degree with [omitted]. Do you know [omitted]? Researcher: Yes, when I entered the laboratory, she was already leaving. Interviewee: I would recommend you [omitted] and [omitted].

6 Quotations:

2:10 ¶ 49 in P2 INTERVIEW TRANSCRIPT

We received feedback from users on our forums.

3 Codes:

- _Mechanisms / ● Forums / ● Requirements change identification

2:11 ¶ 49 in P2 INTERVIEW TRANSCRIPT

We had an environment to organize. We had a kind of library of requirements to store and prioritize. We used [omitted], an open-source task management system, but we also adapted it to link requirements to the task and have a way to prioritize, schedule, and evaluate.

3 Codes:

- _Mechanisms / ● Requirements change identification / ● User feedback tools

2:12 ¶ 57 in P2 INTERVIEW TRANSCRIPT

There were a lot of posts on the forums, and the analysis was manual, yes, but some users preferred to convey their needs in a meeting; they didn't have that incentive to use the forum.

3 Codes:

- _Mechanisms / ● Forums / ● Requirements change identification

2:13 ¶ 93 in P2 INTERVIEW TRANSCRIPT

We received feedback on the forum and in-person and online meetings.

3 Codes:

- _Mechanisms / ● Forums / ● Requirements change identification

2:14 ¶ 101 in P2 INTERVIEW TRANSCRIPT

we communicated through this environment that I told you about, the environment that we adapted for task-oriented requirements management.

3 Codes:

- _Mechanisms / ● Requirements change identification / ● User feedback tools

2:15 ¶ 121 in P2 INTERVIEW TRANSCRIPT

User feedback from a crowd benefits requirements management only if there is a small strategic control group of stakeholders to manage this feedback. Otherwise, I think it doesn't benefit because it becomes disorganized.

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

4 P3 INTERVIEW TRANSCRIPT

Text document, 3 quotations

P3 INTERVIEW TRANSCRIPT Researcher: OK, recording started. [omitted], thank you again for your participation. In this initial part of the interview, I comment on some concepts that will be important for the questions I will ask you later. You may not know some of them in theory, but you should probably know in practice. The first one I will talk about is software ecosystems. Software ecosystems have many definitions in the literature, and I brought the one that I consider the simplest, which has a lot to do with the environment in which you work. We consider a software ecosystem to be a group of developed and co-evolved software projects in the same environment. The idea of an ecosystem comes from the natural ecosystem, where species help each other or compete to survive in their shared environment. I'll show you an example of what a software ecosystem usually looks like. In this example, we have a common technological platform and several software projects that co-evolve consider this platform. This platform has a central organization that controls it, which has internal developers developing the platform and other products. Furthermore, there are external developers from other organizations who develop projects around the platform, and there are also end users who use both the platform and the projects, as well as the software that is developed for that project. OK? Interviewee: OK. Researcher: An example of a software ecosystem is Amazon. Amazon has its central platform where you make purchases, which you probably know, right? However, Amazon nowadays has other software products that have emerged around its main product. For example, Prime is a streaming service, and Music is a platform similar to Spotify and other products, as shown in the figure. These products often have different architectures and are developed by different people. They use APIs to integrate with other products in the ecosystem and the Amazon platform. Interviewee: OK, there's still AWS, right? Which is their cloud computing service. Researcher: Exactly. He's there too. So, all these products are part of a closed software ecosystem with an owner but evolving together. Another example you are undoubtedly familiar with is the one I will show in the following image, which are the [omitted] of the [omitted], the [omitted], the [omitted] and the [omitted]. Interviewee: I know. Researcher: I have a central organization, [omitted], which coordinates and owns this platform. I have internal developers, people from [omitted] who develop products in the context of [omitted], and external developers who often work in this ecosystem. Interviewee: I will use this information to contextualize how the [omitted] work here at [omitted]. To contextualize them in a software ecosystem. Why did I interrupt you? Because I think it's the right time for us to understand the concepts and for me to check if what I'm understanding is really the concept you're presenting. To put it into context, the person who developed the [omitted] was [omitted]. Researcher: Right! Interviewee: Very good. The [omitted] today to give you an idea, I will provide you with some numbers, but later, if you need these numbers precisely, I can give you their source. Researcher: Right, I don't think it will be necessary. Interviewee: [Omitted] Today, there are three main products: [Omitted], which is academic control; [Omitted], which is administrative; and [Omitted], which is human resources. But to support these three main products, there is also [omitted], which is a manager and an administrator, which has a set of features that help these three products in their development about permissions registration, apart from the administrative structure, communication with user, profile, and password. Parallel to this, other systems exist, such as [omitted] and [omitted]. These

products also interact with others. [Omitted] oversees the institution's selection process, including the entrance exam. We also have the [omitted] part, [omitted] is our point manager. In other words, we have other peripheral software products at this center. And today, how are we at the level of development? Today, around 90 institutions in the country use [omitted], both at federal, state, and municipal government levels. This depends on your area of activity. For example, you can have here in [omitted] o [omitted], [omitted], [omitted]. You may have [omitted] a state body that you use. The academic one for high school, the other for higher education, and the other for postgraduate studies. So it is in this context that [omitted] form this ecosystem with different organizations participating in the process of creating these systems. Researcher: Perfect. Interviewee: Today, the person who develops 80% of the code is [omitted]. So maybe I don't know what the job is like. It would be interesting for you to talk to someone from [omitted]. Researcher: Of course. At the end of the interview, I will ask you for recommendations from people who could help me with the interviews. Interviewee: I think that at some point, I will be limited by the degree of knowledge and importance of the process of creating products and their requirements. Do you understand because many of them are made by [omitted]? Researcher: Yes, I understand, but what really interests me in your interview is your point of view in the context of the organization where you work, your actual position in the ecosystem, how you communicate with [omitted], how you send requirements there. How do you deal with the crowd in your context, and how do you consider yourselves about the multitude of other organizations that make up the ecosystem? Interviewee: In this context, I see [omitted] as a partner unit of [omitted] in the creation process. How we contribute to the ecosystem. We contribute with some things, not many, but we have the distinction of being from [omitted]. For example, in the process of creating notes, the business rules and requirements of [omitted] are different from others. So, here comes our difference. Today, there is also a strong focus on digital diploma projects. The federal government established that everyone must have a digital diploma implemented in institutions two years from now. Part of the concept's development was based on discussions in working groups, which are the ecosystem partners. In a meeting with [omitted], a committee was set up to discuss these needs and seek to meet the legal requirements defined by the government. So, we also have this development side using the partnerships formed in the ecosystem. Researcher: Right! Interviewee: [omitted] certainly has its external developers, with the implementation of some service in partnership with other institutions and systems. Researcher: Note that in an ecosystem, the organization's perspective will depend on its position in the ecosystem. For example, if I am interviewing someone from [omitted], these people often do not have the most holistic view of the ecosystem. However, if I were interviewing someone at [omitted], they might have a broader view of external developers. That's important, and that's what I'm looking for. So, we are aligned on the concept of software ecosystems. Interviewee: Yes, yes, let's go. Researcher: Regarding requirements management, the most common concept of requirements management that we can find in the literature is the one that relates it to changes in requirements. This must happen a lot to you and [omitted]. You must request many changes to the requirements for [omitted] so that the systems adapt to your reality. So, when we think about requirements management, this includes how you analyze the impact of a

change, how you manage the changes, how you receive these changes, and what you do with them. There may be some changes that you can resolve yourself, for example, the issue of permissions, the issue of [omitted] that you mentioned, or maybe things that you have to send to [omitted] asking for something to be changed. Perhaps these are things you do yourself, OK? Interviewee: Okay. Researcher: Requirements management involves some related activities, such as requirements communication, negotiation, and prioritization. Negotiation in your context can have several levels. How do you negotiate with management? If and how you negotiate with the multitude of students and teachers and how you negotiate in the context of [omitted], which serves more than 90 institutions, right? If there is a council for this, a public consultation, meetings to decide something, and negotiations on requirements, Interviewee: Yes, precisely that. Researcher: The last concept is related to what we call crowd-based requirements engineering. What would that be? Automated or semi-automated approaches collect feedback from a large set of users or potential users, such as a crowd. A crowd in software ecosystems is a large heterogeneous user base giving requirements on multiple channels with multiple stakeholders. Furthermore, user feedback is an important resource in modern software development. A very simple example is smartphone app stores, like Android or Apple's iPhone, where you rate apps with stars and can enter comments about them. We consider Android a software ecosystem where I have several products that evolve together, right? Interviewee: We consider it, but we have many obstacles not only for us to believe but also to respond satisfactorily. Researcher: This is where this second question comes in: what approaches do you use to identify this requirement? It can be a technique, a method, or a tool. Knowing this difficulty, when you need to know the students' and teachers' opinions, what type of approach do you use? Interviewee: We use questionnaires to survey requirements with this crowd, but I think that these questionnaires fail because the vision of those who respond does not represent the whole. Sometimes, you want to focus everything on a single target, and sometimes, the responder's vision doesn't portray the whole. I'll give you an example, courses are evaluated at [omitted]. Huh? That's cool, let's ask [omitted]. [omitted], what do you think of the structure of the [omitted] library? Wow, but I'm not a user here, but I'm there as a technician. Do you understand me? Then, they will ask specific questions because they made an evaluation form, and I don't have answers. But, I don't know, I have to choose, so I think that at times these surveys fail due to the people they are aimed at, that is, I can't ask something to a person who doesn't know, who doesn't live that. Did you understand? It's as if I were part of a survey, and I'm not best suited to answer. That's what I feel about the tools and techniques that I respond to and participate in. Researcher: They end up not being aimed at the target audience, right? Interviewee: I think there are techniques, but you approach target audiences that will answer that but are not suitable for that research. And then what ends up happening is that you don't have the reflection, the correct result. When tabulating this, you don't have a target audience that will reflect the desires of those who should. Interviewee: Right. Researcher: So, you can rate Facebook with five stars or four stars, and you can also provide feedback and report a bug, but imagine that there are thousands of people giving feedback on an application like Facebook. So, when we have a lot of users providing feedback, it becomes difficult to interview someone or difficult to use more traditional requirements

engineering techniques. In these cases, we need something more automated. For example, today, we can do these things with the help of artificial intelligence. Artificial intelligence projects evaluate all these comments and provide feedback on what is being mentioned most and what is the most recurring bug, and this helps us, for example, to identify a change in requirements. Interviewee: Right. Researcher: Okay, so let's move on to the next part of the interview. We now focus on online feedback from users in a crowd. Have you ever considered a large and heterogeneous set of people, what we call a crowd, to identify requirements or changes to requirements? You just said that you need to listen to the student community now. It would be best if you listened to the teacher community. So, have you considered this large community to be able to identify a requirement for the systems involved in [omitted], for example? Interviewee: I can say what we perceive. This is a perception as an analyst of the institution, as a servant. Today, we have a difficult time accessing this crowd and, so to speak, talking and brainstorming to look for solutions or identify problems. So, what I notice in our institution is that our institution is significant, and somehow, you have jobs that work in isolation. People do not have a cross-sectional view of requests for changes to [omitted], information. People really want to solve their problems, but they forget how this impacts other actors or systems in the ecosystem. So, what do I notice? I realize that every member of the community sees his side and doesn't care about the complementary member, doesn't care about another community, doesn't care about the external user, doesn't care about who will develop, doesn't care about the management, doesn't worry about operations. I miss that communication. The crowd has its demands; it identifies itself but does not interact to seek improvement. The members of this crowd do not integrate. They do not talk to each other. Researcher: This is an interesting point. Perhaps this was one of the motivations for the emergence of this area. For example, in the example of the app store on smartphones, today, everything has become an app, right? Everything was in the palm of your hand. What happened? The applications started to have a large number of users, and it is very difficult for you to serve this crowd, whether because you can no longer interview or brainstorm with a representative set of stakeholders because it is difficult for you to gather, it is difficult for you to grasp the concept of all. So, the idea of considering crowd feedback involves providing several feedback channels for users to evaluate applications. That's why today, when you make a purchase, it's so important that you evaluate the use of the application, because through that evaluation, that feedback, you can have a general analysis, a global analysis to know what's happening in the application. However, doing this manually is difficult. Therefore, today, more automated techniques exist, such as repository mining and artificial intelligence algorithms. So, the question is along these lines. I believe you somehow consider crowd feedback to identify requirements, correct? Researcher: Great, I'll move on to the next question. It's the same thing here, but now we don't want to identify requirements but rather prioritize them. Let's imagine that you have several demands from teachers, students, management, and the government, in short, several demands related to the requirements of [omitted]. How do you usually prioritize these demands? Prioritize these requirements. Do you consider the opinion of the crowd to prioritize? For example, do you think what students want most to make that a priority? Or is the priority created by management or the government? How do you usually prioritize

requirements with these changes in the context of the ecosystem? Interviewee: The priority comes from management. We are subordinate to a board. What's the problem with that? The problem is that the councils rarely meet and decide on daily technology agendas. In other words, within our organizational structure, we are subordinate to a committee that meets every six months to decide on a plan. So, my criticism regarding this is about the structure, but I can't change that. I can't help but express my opinion and my desire. So, there is no point in receiving demands from the community, going through the analysis of a council that takes six months to meet. To decide on technologies that we do not expect. Researcher: Now imagine that you have, let's say ten demands and need to send them to the council, but you don't send 10, you will send five. How do you prioritize this? Do you consider the crowd? Interviewee: We use the criticality matrix. We carry out this criticality and risk analysis. We analyze everything using this criticality matrix. Researcher: Based on this, do you arrive at a priority? Interviewee: Exactly, we prioritized based on this matrix, but as a suggestion. The definition belongs to management. Researcher: Got it! I will move on to the next question, which is also related to what we just talked about. As you use a criticality matrix based on the demands that come to you, some of these demands may not be technically viable. How do you carry out analysis, and why can some have a very large impact, which can make it impossible to use the system or generate bigger problems? How do you negotiate these demands? And with whom? Interviewee: We have the nature of demand within the criticality matrix. So you have legal enforcement stuff. Legal enforcement, at some point, enters prioritization status. I don't have much choice. Now, we also have requests from the academic and general communities. What impact will this have on resolving that demand? And this impact will give us the question of urgency or non-urgency. We also have to look at the available resources. So, there are a series of factors. Do we have resources available? Sometimes, you want to meet that demand, but we lack the resources. So, when we create this criticality matrix, we try to respond to these criteria, right? And through the criticality matrix, an index that gives a signal is generated, right? We use the criticality matrix to provide an index, and based on this index, management can choose. Researcher: Right. How do negotiations occur within the forum with other institutions in the ecosystem? Interviewee: Let's separate it then; our internal demands have an impact; we classify them, and management will decide. For example, if management understands that there is no impact on demand, there is not much we can do. Researcher: How do external demands work? I also think it's interesting that we have this view. Interviewee: The external is like this. For example, if you come with an imposing demand in terms of a deadline, we try to make it viable. But sometimes the deadlines don't match the efforts and available resources you have. There is no point in determining that we will have two years to implement the digital certificate if we do not have the necessary infrastructure or if the equipment has not been purchased. Researcher: I was very curious to know how the negotiation of requirements works with other cooperation institutions, or rather the ecosystem. Could you tell me more about this? Interviewee: I'll give you an example. We had a problem with the login of international students who didn't have [omitted]. This problem affects all institutions, but we were the key. We asked what it would be like, and they said they were still looking for information from the government, and we said we already had the solution. We implemented the solution our way and

modified it to meet this scenario. Nobody in the corporation had thought about that.

Researcher: How do you usually receive this feedback? Whether from students, teachers, or this external community. What tool do you use? In other words, online tools, email, telephone, understand? Requirements changes, change requests. How do you usually receive demands from this number of people? You mentioned the form several people can send you a while ago. Would that be one way? Are there others?

Interviewee: In reality, it is like that. As [omitted], we are not on the front line to be directly demanded by the community. In other words, we receive requests for problem resolution, but in our understanding, it should not be the input channel. The entry channel in our vision must be through the deans and the bodies in direct contact with the community. So, suggestions for improvement from the crowd must come from the pro-rectors.

Researcher: It's as if the pro rectorate were the intermediaries. They listen to the community, define some requirements, and come to you with something more defined. And does this process usually take place via email?

Interviewee: Yes, we are not on the front line. Of course, sometimes these demands also come from management, which defines some things that need to be improved [omitted].

Researcher: And don't these demands arise through other means of communication as well?

Interviewee: They even appear, but this is not the official process.

Researcher: The next question is whether you know or have used automated or semi-automated approaches to manage requirements in your context. In the context of the ecosystem?

Interviewee: I don't know. I have already developed a little robot for report analysis, but not as a requirement. I even thought about making one for us here. But I still see some limits due to our service and how we enter our service level. We need to have a single platform. Today, it is pretty diverse in terms of our demands. Our one only serves one part, the hardware part, but it doesn't serve people who work with software.

Researcher: The last question is whether you consider that listening to this crowd would benefit your work, then managing these demands, which are requirements for you, in general, are the demands for change. Do you think listening to and understanding this crowd benefits requirements management in a software ecosystem context?

Interviewee: Of course, as I said, our systems are traditional, but they are conventional and need to evolve, and there is nothing better than these tools that will help you in decision-making, in the search, in the response, in the feeling, in what makes it difficult. For example, right when the pandemic hit, we needed to implement internet assistance for students. So, how do you build this functionality? How to get it to students' cell phones? So, it would be best if you had this feedback not only for improving the service but for building, let's say, innovation, things that often didn't cross the analyst's radar but are the reality of users out there. It's the reality of that student who often doesn't have the means to express it. A severe problem we have is updating registrations. I think most systems are problematic because you take the basis of one data; you take the basis of another. There is no standardization of business or standard definition of concepts, such as what a person with an economic class is.

Researcher: Thank you very much, [omitted]. We have reached the end of our interview. I thank you very much for your contribution. I'll email you later to see if you have someone to recommend. All good?

Interviewee: Yes. Okay, you can send it.

Researcher: Good afternoon, let's talk.

3 Quotations:

4:1 ¶ 62 in P3 INTERVIEW TRANSCRIPT

We use questionnaires to survey requirements with this crowd, but I think that these questionnaires fail because the vision of those who respond does not represent the whole.

3 Codes:

- _Mechanisms / ● Questionnaires / ● Requirements change identification

4:4 ¶ 94 in P3 INTERVIEW TRANSCRIPT

when we create this criticality matrix, we try to respond to these criteria, right? And through the criticality matrix, an index that gives a signal is generated, right? We use the criticality matrix to provide an index, and based on this index, management can choose.

3 Codes:

- _Approaches / ● Multi-criteria decision-making methods / ● Requirements prioritization

4:5 ¶ 127 in P3 INTERVIEW TRANSCRIPT

Of course, as I said, our systems are traditional, but they are conventional and need to evolve, and there is nothing better than these tools that will help you in decision-making, in the search, in the response, in the feeling, in what makes it difficult

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

5 P4 INTERVIEW TRANSCRIPT

Text document, 8 quotations

P4 INTERVIEW TRANSCRIPT Researcher: Okay, the recording has started, right? Let's now begin the first part of the interview. In this part, I explain some important concepts so you can answer the questions I will ask you later. First, let's talk about software ecosystems. I brought up a concept that I consider the simplest so that the different people I am interviewing can understand this concept. Often, these people are contributing to a software ecosystem, but they don't know the definition of it. A software ecosystem can be defined as groups of projects, groups of software projects that are developed and co-evolved in the same environment. To illustrate this concept, I brought some examples. Software ecosystems have a common technological platform, which is software that offers services for these other software projects to co-evolve. That is, it often enables integration and allows these projects to benefit in some way. Furthermore, in an ecosystem, there is a diverse set of actors, such as the central organization, which is the owner or responsible for the platform, and the internal developers of that organization, as I made clear in the image. Some external developers often develop these projects around the central platform. There

are also end users who benefit from the functionalities of all products in the ecosystem. An example that I bring of a software ecosystem is that of Amazon, many people know as the Amazon e-commerce environment, which has evolved into several other products that are around it, such as Prime Video, which is a streaming service, or AWS, which is a cloud service and other services. Another example I'm giving is [omitted] of [omitted], a group of software projects that co-evolve in the same environment with different integrations and many stakeholders involved. Right? Interviewee: Okay. Everything understood. Researcher: The second concept is requirements management; the concept of requirements management that I have been investigating is more related to a process that accompanies the life cycle of a requirement, focusing mainly on the changes that occur in these requirements in the development of these projects that are around the platform or in the ecosystem. So, when we talk about requirements management here, we must be concerned with some specific activities, such as communicating requirements between the organizations cooperating around this ecosystem. How do changes occur? How are they controlled? How do requirements negotiation occur between these partners? What about prioritization and traceability? These are some activities that the literature, in general, relates to requirements management. All good? Interviewee: Yes. Researcher: The third concept is slightly newer in the context of requirements engineering, which we call crowd-based requirements engineering. This concept emerged in 2015 and became better known in 2017. Crowd-based requirements engineering uses automated or semi-automated approaches to collect and analyze a large set of information that we can call user feedback or a new set of several users, which would be the crowd so that we can carry out the requirements activities. A crowd in software ecosystems is a large heterogeneous user base that gives requirements on multiple channels with multiple stakeholders. User feedback is an important resource in modern software development. So, an example I present about this is a smartphone's application store, where you have several applications and can rate them using stars and leave comments that may be related to the product, integrations of that product, and the platform, right? Interviewee: Yes. Researcher: A popular application can have hundreds or thousands of user comments, and very important information is often lost because no one will look at these comments individually. You don't usually have a team to do this; otherwise, it's all very dynamic. To assess whether that comment is insulting or something good. So, crowd-based requirements engineering considers this crowd to generate inputs for requirements engineering, okay? Interviewee: Everything understood, and yes, I do. Researcher: Right. Now, we go to the second part, which is the part of questions related to the concepts I presented to you previously. Any questions regarding the concepts? Are there any questions left? All concepts understood? Do you think you have the experience to talk about them? Interviewee: No, I do have experience. I found it interesting because this concept has everything to do with the software ecosystem, right? Crowd-based requirements engineering. Researcher: The first question I ask you can be related to different positions in ecosystems, in two positions, to be more specific. The first is your position as manager in your organization, being the person who receives the demands from within [omitted], which has a large set of users for the systems you have, the crowd, so, in this scenario, you receive the crowd's demand. However, in the ecosystem, you can also

be in the position of a member of the crowd, the one who makes demands on [omitted] about the systems, understand? So the first question is, if you've ever considered a large set of people to identify requirements or changes to requirements, how would I do? Interviewee: Generally, these changes reached the level of vice-rectors. And these requirement changes, for the most part, were related to [omitted]. So they generally arrived via meetings and some via process, through the protocol module. These changes to requirements were mostly related to [omitted] and generally arrived via email. And then, based on these demands, we met with this demander, and we decided what was going to be done and what alternatives were possible to implement. Then, we defined what to do. Researcher: From my understanding, the deans work more with the crowd than you do. In other words, the students first arrive at the dean's office and from several things that the dean's office receives, both from students and professors, possibly, and then they send you something that reached them, and they could not do so. Resolve, is that it? Interviewee: Yes, but I don't know if the changes that arrived were so related to the crowd, I believe that these changes came because of [omitted]'s own regulations, and not just at the request of students. Researcher: But what about bugs, for example, which could sometimes exist in [omitted], for example, teachers who detected bugs and spoke to [omitted], so it could be that 10 or 15 teachers detected the same bug at the end of the semester, for example, and then they fill someone's email there, in this context of a multitude of users, how did it happen? Interviewee: I understand, yes, that existed, it still exists today. The products related to professors' use and demands arrive via professors. And regarding the products that students use, these demands are sent directly to the projector's office. Students report directly to the dean's office. In some situations, from specific professors, but in a way, it must always come through the pro-rector's office, and if it were with postgraduate studies, it would come from the other pro-rector's office. That was the two links [omitted] and [omitted]. And if it was related to [omitted], then it alternated. If the module were related to the Dean of Administration, it would be the Dean of Administration. If it were part of the contract, it would be the people from the contracts area. In [omitted], a lot depends on who uses the unit system. Researcher: Right, thinking now about what you said about the demands being directed to the [omitted], the [omitted], the [omitted], and that could come from different vice-rectors or even the rectory or other sectors of the university. This is already a large group of people because then you have pro-rectors involved. So, you already have a more extensive set. How do you prioritize these requirements or changes that come to you? Interviewee: Prioritization was defined in the digital governance committee. There were representatives from the deans and the IT area. And as our team is small, they received demands from the systems and these demands were prioritized and the president of this digital governance committee, if I'm not mistaken was the vice-rector, he defined what the governance committee's priority was digital. Researcher: Got it. And then, these priorities were sent to [omitted], you from IT passed them on to [omitted] is that it? If they weren't made by you, for example. Interviewee: It depends. If they were specific things of ours, we would act on the demand to meet our needs. Now, if it were an issue, for example, a change in legislation involving all cooperative members because the bill is the same for everyone regarding human resources, assets, and bidding. Then we sent it directly to [omitted] and when we forwarded it to them, they already knew why it was a

general demand, we just made it official with [omitted]. Researcher: For example, sometimes you identified a bug, and someone identified it [omitted]. You fixed it for [omitted], but it could be that this bug was rolling out to other universities. What did you do in these cases? Interviewee: [omitted] was a system for sending feedback related to innovation, a new module, and even a new system. In these bug situations, which did not involve changing requirements, we used [omitted]. For these bug issues, there was an unexpected behavior correction request form. We filled out this form with the problem data and screenshots. [Omitted] gave a deadline for resolving the issue. They checked everything and gave us an answer. They have an SLA, but I don't remember their SLA. It was a form for meeting demands and correcting unexpected behavior. Researcher: So, the changes that you didn't make, they had to make, right? You sent some demands for them to do, but there were things that you did yourself. When it was a more specific adaptation for [omitted], correct? Do you know how these changes were prioritized at [omitted]? This thing, for example, about voting that you mentioned there. Interviewee: To define new features. Researcher: And for these cases of adjustments, because, for example, you send demands using this form, and other universities also send demands. Do you know how this happened? This prioritization took a long time. How did it work? Interviewee: I don't know how they met these demands at [omitted], but I know there was a level of criticality when we filled out the form. We had to define the urgency of the demand. It had some levels of criticality. I think it was based on this level of criticality that they prioritized. Researcher: Possibly, they had some indicator to act on the demands, right? Interviewee: Yes, yes. Researcher: And the next question is related to requirements negotiation. Note that many people relate negotiation to the prioritization of requirements, but when I am referring to negotiation, I often refer to conflicting requirements. When I have conflicting requirements, then this requires negotiation. For example, member A wants the color blue for something, but member B wants the color red for that same thing. So, is the central organization going to do blue, or is it going to do red? Will she answer one or the other? She can't meet both, or she will find a compromise through negotiation and will issue a yellow card, understand? Has this ever happened? How did this happen? Interviewee: In fact, this never happened because the changes in requirements involved a particularity of the organization; it was the organization that had to do it, and it was the one that had to make that change. It was the university's team. They provided technical support, clarifying doubts. Look, I want to do this here, how am I going to do it, where am I going to move it? Which class am I going to change? What script can I run on the database? Let's say they only provide intellectual assistance, but there is no conflict because any demand that specifically involves that body, that body's technical team would make. This customization, this change. Researcher: I understand, but I'll give another example to see how it works in this other case. [omitted] will launch a new functionality for one of the systems. So, how does it present this functionality to you? How does she collect your opinion? You said she sometimes launches a form for you to give your opinion. Based on these opinions, she would come back with you to discuss; University A gave opinion X while University B gave opinion Y. From that, we can discuss the best opinion. Based on this, we are presenting you here. Can you ask us to change something? Is there such a moment, too? Interviewee: First, the [omitted] platform was developed to meet the needs of [omitted]. [Omitted] is a

university organized at the level of extension, research, teaching, and administrative procedures involving these activities. They are very organized and very planned. When a demand arose within [omitted], they developed this module and this new functionality and placed it in the reference repository. And if any university wanted to use it, they updated their system to the latest version. So, this is what happens. That's why I'm saying that there were no conflicts because when a university verified that there was a new functionality and that this new functionality would be interesting for that university to use, the first thing they did was the following, they checked if the functionalities were by the internal regulations here. Still, I'm observing how this module was done, and the procedure they adopted here is much better than mine. Then, they could decide to change their regiment. So, conflict generally didn't happen. Researcher: So you consider that the university, the keystone, often sets trends where members change their internal regulations to control the keystone? Interviewee: I'm talking about this in my research. I interviewed ten institutions, two from each region of the country. I'm not going to say 100% because it's dangerous, but 85 to 90% of institutions have adapted and updated their regulations to be able to use the systems. Why did they do this? Firstly, because [omitted]'s procedures are really very good. Secondly, there are many demands in the area of assets, in the warehouse, in the teaching area, and in digital diplomas, which none of these universities can meet in a short period. Then, they meet with the gestures of their respective universities and bodies to address this issue. Universities say they need to meet the demand for digital degrees in at least three years. So, they found that it was better to adapt to the system because they would meet all the demands that the government requested in a short period. They don't want to have problems with the union's audit court or with the union's general controllership. So they prefer to use the systems. One of the managers told me he even received praise from the union's general controller because they implemented the electronic point in the institution. The systems are integrated, the [omitted], [omitted] and [omitted]. Suppose we update, for example, [omitted], to a version. They could fulfill the electronic point request. They had to update the [omitted], update the [omitted], and so on. Researcher: They co-evolve in the same environment, correct? Interviewee: They evolve together, that is, let's stay calm. There is no fine for the Rector or the pro-rector. Do we agree that [omitted]'s procedures are good? We agree. It wasn't just [omitted] that did this. 90% of the universities in cooperation did it this way; that is, today, they are in a very calm situation, with few IT servers, but meeting all the needs that the federal government has today to request for the cooperative entities. So, that's it; there is no conflict. For example, many universities have customized their systems, but they have customized around 30% or 20%. So, in other words, there is no conflict because those who want to customize and customize with their technical team and those who don't want to don't customize and adapt how it was developed. That's why I say there's no conflict. Researcher: Okay, let's go to the next question. When do you receive this feedback or send it? In this case, here, thinking, how do you receive internally from [omitted] and how do you send? How do you do it? What are the tools and communication channels? How does communication of requirements occur in this [omitted] and [omitted] environment, in cooperation, in the ecosystem, understand? Interviewee: It is precisely with updating these tools that [omitted] has, which is [omitted]. First, they consult the tool [omitted] to clarify

business rule issues and check whether it meets the institution's business needs. We always look for [omitted] to see how many people have experienced the same problem. Otherwise, you have two actions. The first action is to request the change from [omitted] through [omitted], and the second is to prioritize and negotiate with the institution to make these changes. Researcher: There are also questionnaire for bugs, correct? Interviewee: Yes, there is a questionnaire for correcting unexpected behavior. Few universities customize the system because there is a risk that they will not be able to update in the future. After all, if this customization is very deep and robust, they may be unable to update to the latest versions of [omitted]. Researcher: With each update, they have to insert their modifications again. Interviewee: That's right, there's the issue of merges in Git. When we update the versions, he says there's that merge issue, right? Comparing the codes, some universities changed the system so much that when it came time to merge, they could not do anymore because there were so many changes they had to include in the new version that it became unfeasible. For example, [omitted] can no longer update the systems, or at least [omitted] cannot because they have already changed the system so much that they can no longer update the reference version of [omitted]. What was their loss? They are very confused about the issue of the digital diploma. Because the diploma will be offered by [omitted], but they cannot update it. Researcher: And in [omitted], you said it is via a process in [omitted] or via email by the pro-rectors themselves, right? Interviewee: That's right. Researcher: The next question is whether you know or have used any automated or non-automated approach to support management in this software ecosystem context. Interviewee: We don't use it, but [omitted] manages these requirements well. They have some tools integrated into their Git, and they do requirements traceability. They definitely have it there, including these commits and their traceability with the requirements. They have these tools to do this, but we don't do it, and we don't have the need to do it. Researcher: For example, on GitHub, do developers use issues to discuss problems and exchange information? Do you know if they use Stake Overflow or other tools? Interviewee: Do you want to know if we use these tools? Researcher: The developers at [omitted], for example, these questions you mentioned, this help they give you, for example, how does this happen? When you want to change something. Interviewee: After entering these demands via the form, we have the categories of demands. They contact us via Skype. Doubts are usually clarified via Skype. First, the demand is registered on the form I mentioned. Based on this demand category, someone will contact us via Skype, and then the problem will be resolved using this Skype tool. Researcher: Got it. I will present you with a scenario and then ask, often on GitHub, about these ecosystems that exist within GitHub, NPM, and some ecosystems that are more open within GitHub. So, the discussion is usually held between the developers, who are users of the libraries there; they go to the issues and post the bugs they find, suggestions for improvement, and everything else. But what happens there, what I told you about the app stores, are thousands of conversations, hundreds of thousands, and sometimes the central organization or someone interested in solving that bug cannot easily find or cannot know what harms that ecosystem the most. So, do we have it today? Some automated approaches use artificial intelligence, algorithms aimed at text analysis, analyzing what everyone said, putting it in a vector, analyzing everything, and then saying, look according to the AI and what it learned. We think

that This may be the trend of change in the ecosystem. What is being adopted the most is this resource. Do you consider whether this benefits, in this case, requirements management in the context of your ecosystem where you are involved? Interviewee: I believe that user feedback benefits because it allows ecosystem members to interact. For example, I can access a release from a member, find something interesting, and bring this requirement to my fork as well. In other words, I'm not going to go there and fix a bug of theirs because it's not interesting to do that, but I can go there and get some release from them, an interesting thing, and bring it to my fork too. It can be done, okay? Researcher: So, [omitted]. We have ended the interview, and I thank you very much for your time; you helped me a lot. Would you have someone you could recommend carrying out this interview?

8 Quotations:

5:1 ¶ 25 in P4 INTERVIEW TRANSCRIPT

they generally arrived via meetings and some via process, through the protocol module.

3 Codes:

- _Mechanisms / ● Built-in user feedback modules / ● Requirements change identification

5:2 ¶ 25 in P4 INTERVIEW TRANSCRIPT

These changes to requirements were mostly related to [omitted] and generally arrived via email.

3 Codes:

- _Mechanisms / ● Mailing lists / ● Requirements change identification

5:13 ¶ 46 in P4 INTERVIEW TRANSCRIPT

[omitted] was a system for sending feedback related to innovation, a new module, and even a new system. In these bug situations, which did not involve changing requirements, we used [omitted].

4 Codes:

- _Mechanisms / ● Issue trackers / ● Requirements change identification / ● User feedback tools

5:14 ¶ 46 in P4 INTERVIEW TRANSCRIPT

For these bug issues, there was an unexpected behavior correction request form. We filled out this form with the problem data and screenshots. [Omitted] gave a deadline for resolving the issue. They checked everything and gave us an answer.

3 Codes:

- _Mechanisms / ● Questionnaires / ● Requirements change identification

5:15 ¶ 79 in P4 INTERVIEW TRANSCRIPT

they consult the tool [omitted] to clarify business rule issues and check whether it meets the institution's business needs. We always look for [omitted] to see how many people have experienced the same problem

3 Codes:

- _Mechanisms / ● Forums / ● Requirements change identification

5:16 ¶ 79 in P4 INTERVIEW TRANSCRIPT

The first action is to request the change from [omitted] through [omitted], and the second is to prioritize and negotiate with the institution to make these changes.

5 Codes:

- _Mechanisms / ● Forums / ● Requirements negotiation / ● Requirements prioritization / ● User feedback tools

5:17 ¶ 84 in P4 INTERVIEW TRANSCRIPT

Yes, there is a questionnaire for correcting unexpected behavior.

3 Codes:

- _Mechanisms / ● Questionnaires / ● Requirements change identification

5:18 ¶ 108 in P4 INTERVIEW TRANSCRIPT

I believe that user feedback benefits because it allows ecosystem members to interact. For example, I can access a release from a member, find something interesting, and bring this requirement to my fork as well. In other words, I'm not going to go there and fix a bug of theirs because it's not interesting to do that, but I can go there and get some release from them, an interesting thing, and bring it to my fork too. It can be done, okay?

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

6 P5 INTERVIEW TRANSCRIPT

Text document, 6 quotations

P5 INTERVIEW TRANSCRIPT Researcher: OK, now it's recording. I will start the first part of the interview by introducing you to some important concepts that will be part of the questions I'll ask you next. OK? Interviewee: Yes. Researcher: The first of these is software ecosystems. Do you know this concept? Interviewee: No, I don't know. Researcher: Yes, this has happened to other interviewees too. So, software ecosystems are groups of software projects or systems developed and co-evolved in the same environment; that is, they have integration; despite being independent, one can affect the other and everything else, so when we have this broader analysis. We

call this a software ecosystem. I'll give you a general example of a software ecosystem. Software ecosystems have a common technological platform that brings these projects together. These projects are around this platform. Some actors are involved in this ecosystem. A central organization that owns the platform. Internal developers are the employees of this organization and external developers are people outside of this organization who contribute to these projects and may even own them. The end users benefit from both the platform and the projects. It's OK? Interviewee: OK. Researcher: A practical example so we don't have any doubts is Amazon. I don't know if you know it or have used it. Interviewee: I heard about it. Researcher: Right! Amazon started as an e-commerce company and evolved into a series of products, such as Prime Video, which is Netflix-like streaming. AWS is a cloud service like Google Drive, Dropbox, and other products that revolve around the Amazon platform, so we can consider this a softwareOKsystem, okay? Interviewee: Ok. I watch Prime, and I've already used it. Researcher: This happens in software ecosystems; often, you develop or are a product user but have no idea that it belongs to an ecosystem. You can use just that product, but you can also use other products through integrations. Interviewee: Right, I understand. Researcher: Another example of a software ecosystem is [omitted] from [omitted]. You obviously know this, right? Interviewee: Yes, I had already made a note about this. When I saw the concept of central organization here, I put it at [omitted] and external developers at [omitted]. Researcher: Exactly. I had even adjusted this point in the image. When we see it in the context of [omitted], some people may see [omitted] as the central organization, but if we have a broader view, we analyze the ecosystem. OK? Interviewee: Yes, the context of [omitted] itself. Researcher: The following concept is requirements management. It's a concept we usually learn in college that is related to requirements. In general, we are demand-driven, right? Demands arrive and often become requirements for the software. When discussing managing requirements, we usually talk about changing requirements or even new ones that will impact the work. For example, in the context of [omitted], you have teachers who demand changes or new functionality, you have students who request this too, you have management, you have a government, and you have to adjust, managing all of this to be able to be something feasible to achieve. Interviewee: Exactly. Researcher: The third concept is crowd-based requirements engineering. Today, we are experiencing a significant increase in the use of mobile device applications. So, a popular app like WhatsApp or Facebook has millions or thousands of users, and sometimes you can think of a new feature. You shouldn't just think about a few of these users but rather the feedback from many users. User feedback is an important resource in modern software development, often containing requirements that help address user concerns and desires for one or more software products. A crowd in software ecosystems is a large heterogeneous user base giving requirements on multiple channels with multiple stakeholders. Crowd-based requirements engineering includes all approaches that engage a crowd of primarily unknown people to perform required engineering tasks or provide requirements-relevant information. Let's think today for [omitted], think about a new functionality, we have the primary client, the government, which can bring more imposing demands. However, there is a network with many people involved, so we often need to collect feedback from this crowd. Have you ever seen the app stores where you can start and write a review about the apps? Interviewee:

Yes, yes, I love it. I love seeing the comments. Researcher: One of the goals of crowd-based requirements engineering is to provide automated tools and approaches using, for example, artificial intelligence algorithms so that we can analyze this large amount of data that came from user feedback. Based on the concepts presented, do you consider that you have experience and are involved in the context of my research? On the form, you said you didn't know. Interviewee: Yes, I didn't know the concepts, but after your explanation and by bringing the example of the systems I work with, I am inserted in this context. Researcher: Now, we start the question part. Have you ever considered this large number of people that we are calling a crowd to identify requirements or changes to requirements? For example, in your ecosystem have you ever noticed many people complaining about something or requesting a change and from that you identified changes in requirements? Interviewee: We access [omitted] to see how many people have experienced the same problem and how they resolved it. All members could make suggestions on the forum. We checked how many tried to solve it and how they did it, considering the different versions of the systems. If I can't get a response from [omitted], we open a ticket to [omitted] using another specific tool to report unexpected system behavior. In this tool, we open what we call a technical ticket. Researcher: I ask you, what kind of approach do you use to identify these requests from this crowd of students, teachers, and everything else? Interviewee: We use mailing list, and it worries me when several requests start to arrive regarding the same item. Researcher: Have you done any analysis of this large amount of data? Or is the analysis completely manual? Interviewee: Generally, they are more on our mailing list. Researcher: And have you considered this large number of people when you have to prioritize these or changes in requirements? As you mentioned, when a demand arrives for you, how do you prioritize this demand? Interviewee: When I receive several demands, for example, I arrived today and opened the system, and there are ten processes, so I visualize the subject of each one, and depending on what I can resolve more quickly, I prioritize it. I'll cite an example: a permission process to access the [omitted] arrived here; how quickly can I do it? The user wants permission to create an email. It's not in my department, so I'll forward it to another department. The user needs a modification within the system. This will require more things, and I will leave it on standby. What I can resolve immediately, I prioritize. What I can't do, I'll go for later. Researcher: And how does it happen when demands affect superior management, be it the management of [omitted] or the management of [omitted] itself? How do these demands work? Interviewee: We work under the term business analyst. Depending on the demand, I first consult the technical analyst. And then, depending on his opinion, a manager will stop giving [omitted], and depending on that manager's position, it will go to the governance committee. Researcher: Right. When do you get to [omitted], for example? You know. Interviewee: There is a tool to register these demands related to system customizations. This system defines the demand, its formalization, and the priority and status of each one. Researcher: The following question concerns negotiating requirements. Do you usually negotiate these demands? Whether, for example, with the people who asked, with the managers, or with a committee with [omitted]? Is there a committee that has a discussion? How are these demands negotiated? Interviewee: I don't usually participate in negotiations with other institutions in the network. I participate more in the internal part of

[omitted]. Researcher: How do negotiations work internally with your boss and managers? Do you usually involve users from a crowd? Interviewee: As I said, depending on the type of demand, we make a note as if it were a process [omitted]. When we register the note, and I enter the day it was made and what depends on it to resolve that demand. Researcher: How do you get this user feedback from a crowd? You already mentioned email. How does this communication occur? You already said you have a note formalized in [omitted]. Interviewee: Yes, all demands are formalized in [omitted]. Researcher: Did you also mention that this communication occurs outside the [omitted]? When you use [omitted], how does it work? Do you contribute? Interviewee: I still don't contribute. I have only received contributions there. Researcher: And the people who usually respond? For example, are they more from [omitted], or have you already had a response from someone at another institution? Interviewee: Within [omitted], I receive more from other institutions than from [omitted]. Researcher: Interesting, then, often the [omitted] community survives on the cooperation of other institutions. It's not just the [omitted] people who respond, right? Interviewee: No. Researcher: The next question is, have you ever used any approach, whether automated or semi-automated, to manage these demands that you receive daily? Interviewee: No, actually, as I said, I organize things my way. I'm classifying and organizing them within our systems. Researcher: Do you consider that taking user feedback from a crowd into account benefits or hinders your work? In your work, does listening to many people, even considering the filters carried out by the pro-rectors, benefit or hinder you? Interviewee: I believe that if this feedback were focused only on some sectors of the [omitted], such as [omitted], it would be more beneficial. Researcher: Now let's think outside of [omitted]. Do you consider feedback from other ecosystem partners registered with [omitted] helps or hinders your work? Interviewee: Given the experience I've had to date with [omitted], feedback helps. There hasn't been one, yet that couldn't be resolved with the help of [omitted]. To date, my experience at [omitted] has been excellent. It helps me solve problems and identify others. Researcher: We have reached the end of the interview; I would like to thank you for your time. I will stop the recording and ask if you have someone you can recommend so that I can interview, please let me know.

6 Quotations:

6:8 ¶ 48 in P5 INTERVIEW TRANSCRIPT

We access [omitted] to see how many people have experienced the same problem and how they resolved it. All members could make suggestions on the forum.

3 Codes:

● _Mechanisms / ● Forums / ● Requirements change identification

6:9 ¶ 48 in P5 INTERVIEW TRANSCRIPT

If I can't get a response from [omitted], we open a ticket to [omitted] using another specific tool to report unexpected system behavior. In this tool, we open what we call a technical ticket.

3 Codes:

- _Mechanisms / ● Issue trackers / ● Requirements change identification

6:10 ¶ 52 in P5 INTERVIEW TRANSCRIPT

We use mailing list, and it worries me when several requests start to arrive regarding the same item.

3 Codes:

- _Mechanisms / ● Mailing lists / ● Requirements change identification

6:11 ¶ 56 in P5 INTERVIEW TRANSCRIPT

Generally, they are more on our mailing list.

3 Codes:

- _Mechanisms / ● Mailing lists / ● Requirements change identification

6:12 ¶ 88 in P5 INTERVIEW TRANSCRIPT

Within [omitted], I receive more from other institutions than from [omitted].

3 Codes:

- _Mechanisms / ● Forums / ● Requirements change identification

6:13 ¶ 100 in P5 INTERVIEW TRANSCRIPT

I believe that if this feedback were focused only on some sectors of the [omitted], such as [omitted], it would be more beneficial.

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

8 P6 INTERVIEW TRANSCRIPT

Text document, 3 quotations

P6 INTERVIEW TRANSCRIPT Researcher: The recording starts here. So, let's go. Part one of the interview is where I present some important questions, I will ask you next. The first concept is that of software ecosystems. I brought a concept that I consider the simplest so that the different people I am interviewing in the market can understand this concept. Often, these people are contributing to a software ecosystem, but they don't know the definition of it. A software ecosystem can be defined as a group of projects, a group of software projects that are developed and co-evolved in the same environment. To illustrate this concept, I brought some examples. Software ecosystems have a common technological platform, which is software that offers services for these other software projects to co-evolve. That is, it often enables integration and allows these projects to benefit in some way.

Furthermore, in an ecosystem, there is a diverse set of actors, such as the central organization, which is the owner or responsible for the platform, and the internal developers of that organization, as I made clear in the image. Some external developers often develop these projects around the central platform. OK? Interviewee: Okay. Researcher: The second concept is requirements management. Sometimes, there is confusion in literature and practice because requirements management is often seen as a synonym for requirements engineering. In the literature, requirements engineering is divided into requirements development and requirements management. I am focusing on requirements management for this interview, as that process is more focused on monitoring requirement changes throughout their life cycle. In this way, the activities I am most related to requirements management, despite being a transversal process. It is the communication of requirements from the point of view of identifying changes, negotiation, and prioritization, which are activities more related to user feedback. Interviewee: Ok. Researcher: The third concept is crowd-based requirements engineering. This concept has emerged due to our new reality in software development. I always give the example of the Android and iOS ecosystems, where you have several applications used by millions of people that have different ways of providing feedback, including star ratings and comments. There is a large and diverse set of data, including compliments and complaints from many user feedback. User feedback is an essential resource in modern software development, often containing requirements that help address user concerns and desires for one or more software products. A crowd in software ecosystems is a sizeable heterogeneous user base giving requirements on multiple channels with multiple stakeholders. CrowdRE includes all approaches that engage a crowd of primarily unknown people to perform requirements engineering tasks or provide requirements-relevant information. Interviewee: Okay, I already know these concepts too. Researcher: From now on, I will move on to the questions part. The first one is about crowd feedback. If you've ever considered a crowd to identify requirements or even changes to requirements through bug reports, things like that. Interviewee: Yes, but we must always measure what is also within our reality. First, we want to have a good relationship with our users. We will never tell the person with the money in hand that we won't do it. At the same time, we cannot deceive a person. So, it's that thing, the answer to this will have to be a response like thank you very much for your comment, we will evaluate, but we always have to analyze the business reality within the organization. There is a new product; no team is explicitly allocated for it, so if there are many features that come with it and some that are not even key to the tool, it could remain in the plans, right? For the next version, I don't know how long later, but it's not something that will necessarily be considered for now. It's a differentiator; now, if it's a tool that's booming and worth the investment, that already has a team allocated, then it's often, the features from then on will come, many of them from the crowd. Yes, it is. Again, this largely depends on the audience you want to reach. For me, a crowd you are considering is a tool being distributed to a very large mass of people, right? Maybe the target audience is small, and you don't want to reach a crowd. Then, I would say that sampling is more relevant than population sampling, a research term. Researcher: In this case, the focus is to think about this crowd as thinking about the idea of collective intelligence that comes from this whole crowd, bringing results,

thinking that this crowd will be motivated. That example you gave initially is famous, with many people commenting and getting points. Let's imagine this scenario, and from that, when we think about the idea of a crowd, I at least see it, which is often when you lose control of doing something manually because you are receiving a lot of things. You won't be able to see comment by comment, for example, and then we go to open source again because there you have those heated discussions, through open issues, and then you go there, and there is a large set of problems. Interviewee: Precisely. This crowd, I consider. Researcher: From that, when you considered this crowd, what specific approach did you use? Was it all manual? How was it? Interviewee: Look, generally, this is manual. However, many times, there is also some tool, something like that, that summarizes this for us in some way. Here in the organization, we have a tool to retrieve comments from the app store and summarize them by topic. It is an artificial intelligence tool. This way, we can group these comments by topic. This is an AI that we have. This way, we can take issues by topic and already have a grouping. Again, this depends significantly on how much I want to achieve and how much is possible and viable, right? Sometimes, someone thinks of an excellent idea, but then I say that's impossible in our current reality. For example, there are a lot of bad ideas, too. The biggest problem with the crowd is often the quality of opinion. Now, if this is a pain for anyone, maybe this is something to listen to. Researcher: Did you develop this tool yourself, or was it purchased? Free? Interviewee: No, these things are being developed here in projects. To give you an idea, they create things to put in stores and stuff like that. I worked a lot with software factories. Researcher: Because we in the project are thinking about something similar, we are developing something similar, considering top modeling, automatic analysis of comments, and text analysis; it's good to know that there is too from you. Interviewee: There is, but it is private. Researcher: Yes, yes, exactly. Considering that we have identification, you said that it happens manually, but you also have automatic and semi-automatic ways of doing this. Regarding the issue of prioritizing these demands from which you generate topics, from demands that are repeated or a large number of exciting demands, how does prioritization occur in your context? Interviewee: That's how we see it, according to reality. We use the app store to retrieve comments and then organize them into topics to analyze them. We try to measure the effort to respond to these suggestions and always try to start with the things that are most talked about. So, 400 users commented on this. It's what was commented on the most, but this is a tricky thing. Maybe this will be left for the next version. So, let's go. This one was commented on by 218. This is easy to do, but it depends on the product. If it's a product that I have a team allocated to the product, then calmly, 400, that's it—doing so for the next version. Researcher: Does it depend on the moment? Interviewee: Yes, and the tool's success is mainly because the company will never want to invest more in a solution, but it will still give a large return. Sometimes, often, it can also be a commercial issue. I've seen tools that entered the market timidly and had many downloads. Researcher: From what I understand, this prioritization is a general impact analysis. A question of first analyzing the number of comments, for example. From this, an analysis of time, say, of time about difficulty and then an analysis of actual interest regarding the market for that product, correct? Interviewee: Yes, this is a holistic issue for the organization. Even though we have evangelists, they have a general vision of the

community, and often, the evangelist is active. The ecosystem's own Evangelist will usually end up getting in touch, and we've had situations where people from stores, you know, who work for the company that owns the store, come and send an email to talk and form a partnership. In this sense, we can even become an affiliate of these organizations. Researcher: How does this relationship occur with evangelists and requirements in the ecosystem? Interviewee: I'll give you an example: Apple, we published an application there, and people liked it. In a little while, someone from Apple, an Apple evangelist, will come and exchange an idea with us and say, Man, really cool, what are you doing? Look, I see a lot of downloads; it seems like an excellent proposal. Soon, when you see them, they will propose a partnership, a particular condition, so that we can meet some objective. Or, you know, these things happen too. But, as much as the Evangelist is active and present, he can also, to a certain extent, help to understand a lot of the behavior and expectations of the ecosystem. Still, there are things that sometimes only over time and the use that will be evidenced. Some things are not so striking at first, and only when users use it do they encounter these difficulties. Researcher: The third question in this part is related to requirements negotiation, prioritization, and negotiation, which are close, right? But then I will give an example: when we are talking about negotiation, we are talking about conflict of requirements. They are requirements that either you choose one, or you choose another, there is no way you can meet one first and then the other, which is. In the case of prioritization, in negotiation you will either choose one or the other. How does trading work for you? Do you take into account a crowd when negotiating? Researcher: And when you have two partners with conflicting requirements, you, as a keystone or even an ecosystem orchestrator, will try to intervene or help in this negotiation. Does it happen, too? For example, some partners are considered more important than others. Interviewee: I think this is not a good way to go, although, yes, it is necessary, okay? But I say this because we don't want to make anyone unhappy or uncomfortable, especially if we are talking about expensive values and values. Even that less important partner often contributes absurd revenue to the project. So, we have to think about the following: when it comes to numbers, there are no arguments. There is an excellent use for the crowd or specific people for us to carry out some analysis, behavior, or demands to influence this change. For example, one of the partners said, we want functionality X, and the other said I don't wish to function. Then he comes to both of them and says, this is an exciting business potential. Trying to convince the group that doesn't want it by saying that it has great potential, it can provide a return, and some people wish to it, chasing it. Now, the reciprocal can also be true. I have 5000 or 10000 comments on this application here, and no one is asking for this. Or we only had two comments asking for this, so we reached a point here. Maybe this is it: you never tell the customer who has the pen in their hand that you won't do it, right? Because he is often the one financing the project, you will say the following, Man, we concluded that this might not be the priority at this moment, considering that there is feedback from so many people or there is a competitive situation with another organization and that demands that we focus on that side. So, you direct the client not necessarily through their ambition but through an aligned vision of what is best for the business; there will hardly be a business conflict in the sense that if you do this, it will help one person and harm another, necessarily. It's usually how we reconcile the idea so that it allows both.

Generally, when someone refuses, it's more because they don't want to invest more in it, so sometimes it's more for you to convince the client to open your wallet a little more. Researcher: It's because it will have an impact, right? As much as she doesn't want to be here, sometimes he doesn't think it's going to be a problem or that it's going to make something complicated; it's simply because she can't invest at that moment or doesn't want to for some reason. Interviewee: I already had a boss who said that he would give his wife a developer instead of a diamond ring at the end of the year because it was more expensive. Did you understand? Interviewee: I know a little, but I never got to work much with open-source people, to be quite honest. A user only negotiates if the app is trendy, like a social network or game that impacts millions of people. Otherwise, they don't negotiate. You will do what seems to be more profitable, has a greater return, or will attract more users, then yes, within the objective that you have there. But generally in the negotiation itself, we will not negotiate with the user when it is a crowd, we negotiate with the user when it is a demand requested by the company or a group of people when it is a closed project. Researcher: But sometimes, even in a closed project, you need to negotiate with partners, and these partners can form a large and heterogeneous group of people, right? Interviewee: I thought it was more like negotiating with the user. Now, yes, with partners, yes. Many times, yes, because it has to be in the partner's interest. We also cannot make it so that, for example, our business rules, or even our prioritization, will be contrary to something, for instance, that a partner is defending. Otherwise, you'll burn your partner's film, right? Some partner was talking to the community, I don't know, saying that there would be something like this, and then we didn't comply. It looks ugly, right? But this has to come from alignment. Before negotiating with the community and publicizing things, we don't have to have a transparent alignment between partners to understand how valuable that partnership is, right? Researcher: And what approach do you know and use to negotiate with these partners? Interviewee: Meeting, I call everyone in the room, let's have a conversation, a brainstorm, until we arrive at a prioritization. And there are things that sometimes you will have to say to your partner straight away. If you want more of this, you'll suddenly have to add more of that. I don't know what, why maybe we won't make it. That's the negotiation. There's no way just one side can commit to increasing the scope. If there is no response, there is also something desired, whether or not professionals are doing it, and the time is expensive. Something must also be said that is very important, but a technology professional's hour is costly in terms of value. I think we deserve this value, but this high value, but it's a high value to allocate it to things without alignment, then you end up taking it out on someone, and then you have to roll back the solution. Then I don't know what. Then, it has already damaged a little of the relationship with the person, and the person has already become suspicious. Researcher: How does requirements communication occur in this scenario? In this scenario that we were talking about just now, right? In this idea of users, partners. You mentioned the app store there, so apparently, it's a channel you use where people go there and communicate possible requirements. Do you analyze it somehow and with partners? How does it work? But, anyway, can there be communication channels? What channels do you use? Interviewee: As for applications that are already on the market, it involves analyzing comments in the application store or emails, something like that, and again, as I always say, there are

no arguments against data. So, put prints of comments there, show them, and do market research. We also have people in sales who often start prospecting and investigating behavior. Communication regarding requirements is checked to see if it is in the backlog. If it's in the backlog, it was as expected. Suppose it's not in the product backlog, the next version, or something like that. Then, you will have to conduct a market study, see the feasibility, and check with partners whether it is worth putting this into production. And whether it's worth prioritizing this as well. Researcher: And how does this verification with partners occur? In meetings, really? Is there another way? Is there a tool that allows partners to monitor some activity? Is there anything in this regard? Interviewee: Look, I'll say this to you, the partners who work more on the development and production side of the tool, a DevOps, a JIRA, something like that so that the person can see the evolution of the project and sometimes a GitHub, correct?? To access the code and everything else, you can create an environment there to test when it's a more executive person who invests but suddenly doesn't have technical skills. We usually make executive presentations that involve the person taking care of this more business part. As a product owner, it helps to compose something to show the evolution of the project, show what was developed, what is on the conveyor belt, and things that were not put on the conveyor belt so we can discuss whether it is worth adding or not, so, That's a lot of you working with the information and also with who you're talking to. If you are talking to someone technical or highly technical, touch the code, show it running, etc. If you are talking to someone more Business, show the planned period, the investment. Researcher: Do you know or have used automated or semi-automated approaches to manage requirements, that is, to identify demands and possible changes to accompany these changes? In this software ecosystem context, have you already talked about the tool you use? Are there others? Interviewee: Look, there might be a solution on the market for this. I don't particularly remember any, but there could be some comment analyst, something like that, or some AI that could exist. I confess that this part is not something I follow closely. But I've seen people, even in sales, use it to identify trends in the store, create material for partners, and everything else. So, there is, but I believe they are private, which gives an excellent idea because, suddenly, later on, you can try to sell something like that. I think something like this must already exist. Researcher: The last question is to know if considering this crowd benefits requirements management in software ecosystems. If yes, or if no, why? Interviewee: You have to have a balance point for things. To a certain extent, feedback from a crowd benefits requirements management. You can prioritize things; you can adjust some things on the conveyor belt with more information or improve things that have already been done and go back and strengthen business rules around those requirements. Yes, in that context, yes. Now, you also have to have good criticality because not everything that comes from an idea is good. Sometimes, the user thinks they have a good idea, but many people still clap and think it's a good idea, and it's not, you know? So, it works if you have a criticality and are clear about your business rules and what is expected for that product as an organization or partner organization. Now, you can't also be that person who does everything to please your user and ends up doing things you shouldn't do to please everyone. You could be destroying your product instead of making a better product. Researcher: Thank you very much, [omitted]. We have reached the end of the interview. Thank

you very much for your time. I would also like to know if you have any recommendations for professionals I could interview. Interviewee: Yes, I do; I'll give you some contacts.

3 Quotations:

8:5 ¶ 25 in P6 INTERVIEW TRANSCRIPT

Here in the organization, we have a tool to retrieve comments from the app store and summarize them by topic. It is an artificial intelligence tool. This way, we can group these comments by topic.

3 Codes:

- _Approaches / ● Artificial intelligence techniques / ● Requirements change identification

8:6 ¶ 37 in P6 INTERVIEW TRANSCRIPT

We use the app store to retrieve comments and then organize them into topics to analyze them. We try to measure the effort to respond to these suggestions and always try to start with the things that are most talked about.

3 Codes:

- _Mechanisms / ● App store / ● Requirements change identification

8:7 ¶ 86 in P6 INTERVIEW TRANSCRIPT

To a certain extent, feedback from a crowd benefits requirements management. You can prioritize things; you can adjust some things on the conveyor belt with more information or improve things that have already been done and go back and strengthen business rules around those requirements. Yes, in that context, yes.

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

9 P7 INTERVIEW TRANSCRIPT

Text document, 4 quotations

P7 INTERVIEW TRANSCRIPT Researcher: I started recording now. I will begin the first part of the interview by presenting some important concepts that will be part of the questions I'll ask you next. OK? Interviewee: OK. Researcher: The first concept is of software ecosystems. Software ecosystems are groups of projects developed and co-evolved in the same environment. That is, they have integration; despite being independent, one can affect the other and everything else, so when we have this broader analysis. We call this a software ecosystem. I'll give you a general example of a software ecosystem. Software ecosystems have a common technological platform that brings these projects together. These projects are around this platform. Some actors are involved in this ecosystem. A central organization that

owns the platform. Let's think of internal developers as employees of this organization. External developers are outside of this organization who are directly involved in these projects and can be owners. The end users benefit from both the platform and the projects. For example, [omitted] can be an environment with several software projects that have integration and evolve together. Is he well? Interviewee: Yes. Researcher: In the case of [omitted], there may be outsourced companies that develop the system, it may have partners, it may have [omitted], it may have, finally, other developers and end users, who are citizens, people who use these systems. One example of a software ecosystem is Amazon. Do you know Amazon? Interviewee: Yes, yes. Researcher: Amazon is an ecosystem because it has several products. Amazon has Prime, which is a streaming service. It has AWS and other products. Interviewee: Right. Researcher: The second concept is requirements management. Requirements management is a process that encompasses tasks that keep requirements evolving, such as requirements documentation, negotiation, traceability, prioritization, change control, and communication. It is concerned with following the entire lifecycle of a requirement. OK? Interviewee: Ok. Researcher: The third concept is a more current requirement engineering concept: crowd-based requirements engineering. Crowd-based requirements engineering includes all approaches that engage a crowd of primarily unknown people to perform requirements engineering tasks or provide requirements-relevant information. It is related to user feedback, an important resource in modern software development. It often contains requirements that help address user concerns and desires for one or more software products. A crowd in software ecosystems is a sizeable heterogeneous user base giving requirements on multiple channels with multiple stakeholders. Today, we have products that thousands of people use, so we go to a concept called collective intelligence or collective creativity. Many people who use your system can give you valuable feedback. I don't know if it's ever happened to you where a user you couldn't imagine arrives with a problem or a solution you hadn't thought about. Interviewee: Yes, it happened. Researcher: So, what happens in this new perspective? I will give you an example related to the development of mobile applications. You use several applications, and people can rate them through stars and comments there, right? Interviewee: Yes. Researcher: Let's move to part 2, the questions part. Are there any questions about the concept I gave you? If there is, too, we can remove it. Is that ok? Interviewee: No, no, without a doubt so far. Researcher: In this part we will talk about feedback from users of the systems that you use and that you work with, they are used by users, citizens or by users who are employees of the bank branches, correct? The managers and the employees. Interviewee: Yes, these two options. Researcher: Great. Interviewee: We are a factory, so we do not deal directly with the end consumer. However, when there is a complaint from the end consumer, it reaches us so we can make improvements, but we don't have much contact with the end consumer. Researcher: Right, so my questions, based on what you tell me now, will be more focused on discovering how these complaints come from the end consumer and how they reach you. Do you understand? Interviewee: Yes. Usually, let's assume that you want to take out a loan but there are problems and you go to the branch and there are problems. Then, the manager emails a specific area of the institution. Researcher: So, do you use email or an internal system? Interviewee: It's a system. Here we have a system. Researcher: Can you send messages within the

system? Interviewee: Users open a ticket with a bug report or suggestions for changes that reach our manager. So, we analyze these improvements and implement them in our API. . Researcher: Can I ask a question for [omitted]? Interviewee: That's right. It's general. Researcher: Thousands of [omitted] can open a ticket, and everything will reach their sector, such as their manager. Interviewee: There are some filters of what is urgent and what is not. Suppose, for example, an institution's application has messages that we send to customers informing them of an unavailable function or unregistered user. Then, let's suppose that a message appears to the end user, and they cannot understand it; sometimes, it is not a well-designed sentence, and sometimes, it is not a friendly message. So, he open a ticket, right? Formalizes this issue in the system. This call arrives at the manager, who passes it on through a service. Then, we observe within our API the improvements of this message to make it look prettier and provide a friendlier message. Researcher: This motivates my second question, which would be about prioritizing requirements. As you were talking about this message, these calls will be opened by different local managers who will reach out to a small group of managers. Interviewee: It's by specific area, so I work in consignment, so everything related to consignment comes to us. Researcher: So it goes to your manager, who is a person, and then he passes it on to you? Interviewee: There are several managers. Researcher: How does prioritization occur in the context of this ecosystem? Interviewee: We see the financial impact, what causes the most economic impact, we put out the fire first. Researcher: What is a business restriction for you as a [omitted]? Interviewee: Yes. There are also others. For example, in addition to having a financial impact, we have to follow some [omitted] regulations, some things related to the consumer code that are more emergency. For example, you have to enter the 0800 number for people who are deaf or hard of hearing, and this is mandatory. So we'll take care of that right away. Researcher: So, when legislation has to be complied with, for example, in this case, it has to have a higher priority, this prioritization occurs a lot with this group of managers, or sometimes they involve, for example, there are cases of having a meeting general and that you will get the managers, the agency managers, representative of the whole [omitted] to get together so we can think about things, what needs to be done first some planning. Interviewee: No, it doesn't happen. Researcher: The other question is about requirements negotiation. And then we'll say when you receive two conflicting demands. I don't know if that's the case for you yet. For example, if a customer complains that phrase x is nasty and needs to be changed, and a manager praises the exact phrase, which is very good, you would have to decide whether to keep or change the word. Both things are opposite. And then you need to negotiate, like, let's talk to this person to explain and try to show things. This negotiation occurs between you and your managers or between your managers and these managers. Interviewee: Yes, we see the viability of the impact, but always with the negotiation of what is possible and what is not, there is something. Researcher: And what is the feedback like after you analyze it? Do you go to this sector, and does it define it, or is there still a process for discussing it with the other parties? Interviewee: We see what causes the most impact and will cause the most financial implications for the institution. The manager assesses this risk, and then the developers generally comply. It ends up reaching a consensus, but it's very calm both internally and externally, we have some, when it's a new service, we have some

alignment meetings and then we can measure what will or will not be viable. Then, what is most feasible, within the impact, not only the financial implications, but the management impact and time impact, and we check and meet what best fits the institution's expectations. But it's a bit. If we evaluate it, it's a bit stuck; if you consider it in a public institution, it's a bit stuck, and it's not easy. In a public institution, if you were to evaluate the business side, some things would be easy to resolve. Still, it takes a little longer than usual because it is a matter of public function. This issue of communication between the parties and see what is or is not a priority. So, if you take it in the general context, it's a bit of a delay. I don't know if I made it clear. Researcher: Yes. You make the history of negotiation very clear by analyzing the impact of a direct relationship. How does this feedback, whether from the user at the end or from managers, reach you, how does this communication of requirements occur within your organization, for example. Interviewee: We have more direct contact with the managers. We have daily online alignments. So, there is this direct alignment. Every day, he has the alignment where he places his needs. The manager doesn't understand the business as a whole. He doesn't know the whole system. So, he stated his business needs, and we evaluated them with the technical team because we are more in the business area here. Specifically, we work more on the negotiation part. It's about seeing the need to know the manager's priority after we align with the manager. We align with the technical area to understand the feasibility. If the time will be viable, see the impact of this request. The technical team checks whether it is feasible or not. Researcher: And then you go back and talk to the managers? Interviewee: We try to focus on the technical part. Because the technical part knows the entire system, so we try to make it decide according to what the technical team is asking. We work like an Uber. We stay in midfield. We listen to the technical area, and then we also listen to the business area to see if it is viable. Still, in general, we try to understand the technical part more and then evaluate the impact of this within the business area. This is unfeasible. Is there really a need for this? And then it ends up working out in the end. Researcher: What type of channels do you use to carry out this communication? Did you say it's online? Which is a synchronous meeting tool. What would it be like? Interviewee: We work with Teams. Researcher: In both cases, with the technical team and as managers? Interviewee: Yes. Researcher: And how do top managers communicate with you? Interviewee: We have some tools. It's like opening a ticket of the [omitted]. A manager from a city in the interior makes a request, asks questions, and passes the question on to higher authorities until it arrives. When he came here, my manager opened a technical service so we could evaluate this end-customer complaint. As I said, we don't have contact with the end customer. Still, on some occasions, we receive these complaints, and it's all via the system or all through Teams, everything online, and manager-to-manager communication. We don't communicate with the client, neither do we with agencies. Researcher: Do you already know or have you used any automated or semi-automated approach to carry out this impact analysis to carry out the fieldwork that you do? Have you ever heard or thought about using something based on the demands, for example, throwing the demands into the system and bringing you a response? Is this impact analysis more or less ready for you to analyze and understand? Interviewee: Wow, it would be good if that existed. But we don't use it. Researcher: The last question is whether you consider that user feedback from a

crowd benefits requirements management in software ecosystems. Interviewee: I believe this feedback is beneficial to our managers. Researcher: And thinking not only about the end user but imagine, as you said, a manager from the interior agency sends some demand, and if we got all this feedback from all the managers before reaching you and had an analysis beforehand to reach your manager, in this case too. Do you think it would help to prioritize? Negotiate? Interviewee: Yes, it would make it a lot easier Researcher: [omitted], thank you very much for your time. We have reached the end. Thank you very much.

4 Quotations:

9:1 ¶ 54 in P7 INTERVIEW TRANSCRIPT

Users open a ticket with a bug report or suggestions for changes that reach our manager. So, we analyze these improvements and implement them in our API.

3 Codes:

● _Mechanisms / ● Issue trackers / ● Requirements change identification

9:6 ¶ 63 in P7 INTERVIEW TRANSCRIPT

he open a ticket, right? Formalizes this issue in the system.

3 Codes:

● _Mechanisms / ● Issue trackers / ● Requirements change identification

9:7 ¶ 111 in P7 INTERVIEW TRANSCRIPT

It's like opening a ticket of the [omitted]. A manager from a city in the interior makes a request, asks questions, and passes the question on to higher authorities until it arrives.

3 Codes:

● _Mechanisms / ● Issue trackers / ● Requirements change identification

9:8 ¶ 119 in P7 INTERVIEW TRANSCRIPT

I believe this feedback is beneficial to our managers.

1 Codes:

● Benefits of user feedback from a crowd in requirements management in SECO

10 P8 INTERVIEW TRANSCRIPT

Text document, 11 quotations

P8 INTERVIEW TRANSCRIPT Researcher: I have just started recording, and we will start the interview. The first part of the interview is related to the presentation of concepts that will be important for the second part, the questions part. I start by

talking a little about the concept of software ecosystems. There are several definitions for SECO in the literature; the one I am using is Lungu's. I consider this definition to be more straightforward and a little more technical. Software ecosystems can be viewed as a set of software projects that evolve in the same environment. Right? Indeed, you already know this definition, right? Interviewee: Yes, I know. Researcher: The example I have presented to interviewees is the Amazon ecosystem, where we have a set of integrated projects that evolve around a common technological platform. OK? Interviewee: Ok. Researcher: The second concept, requirements management, is also well known. Requirements management is a requirements engineering process that focuses on monitoring the life cycle of a requirement in one or more software projects. This relates to control activities such as requirements change management, traceability, and version control. Furthermore, it can also be related to activities such as negotiation, prioritization, and communication of requirements, as these activities can also refer to changes in requirements. Interviewee: Okay, understood. Researcher: For this interview, we will focus more on activities related to user feedback. User feedback is an important resource in modern software development, often containing requirements that help address user concerns and desires for one or more software products. As you know, in software ecosystems, multiple actors can form crowds of users. A crowd in SECO can be defined as a large heterogeneous user base giving requirements on multiple channels with multiple stakeholders. So, the third main concept I want to introduce you to is crowd-based requirements engineering. Crowd-based requirements engineering includes all approaches that engage a crowd of mostly unknown people to perform requirements-relevant engineering tasks or provide requirements-relevant information. Interviewee: Okay. Researcher: Did you already know the concepts? Do you have any questions about them? Interviewee: No, no, I already knew these concepts. Researcher: Right. So, now we're going to move on to the second part of the interview, which is the part where I ask you questions related to the concepts we just talked about. You answer me based on your experience in the software ecosystems you've already participated in. OKK? Interviewee: OK. Researcher: The first question is, have you ever considered user feedback from a crowd to identify requirements or changes to requirements in your ecosystem? Interviewee: We analyzed software repositories to identify and prioritize requirements. We implemented some scripts to identify and prioritize the requirements in these repositories because we could not manage everything daily. We received it to give you an idea, a spreadsheet, for example, you received it in [omitted], and then you received all the data you wanted. We had to create scripts to retrieve data to identify and prioritize requirements. So, we started using data mining methods because we could not manage all comments daily. We wanted to prioritize the requirements according to what was most important to users. We went to the app store to do so, but we did not have much confidence in whether we were covering what was needed within the ecosystem. We also used Stack Overflow and sorted the issue by reputation to get a better view of this. From that point on, we started to mine. Because it was the right tool at that time, we couldn't wait for the internal approval process. Sometimes it took a long time so, even though it was a large, private company, until you told someone that it was necessary or someone that it was important for the ecosystem, you know, these things about demonstrating value.

Until that happens, man, your delivery has already passed, and you've already been asked to bring this identification, so I think that's it. We used topic modeling algorithms and word clouds to identify the highest priority requirements [...]. We started using data mining methods because we could not manage all comments daily

Researcher: What approach do you use to identify these requirements with the crowd? Interviewee: The analysis was really ad hoc. We implemented some scripts and generated some visualizations. We used the social network analysis tool. We used topic modeling algorithms and word clouds to identify the highest priority requirements. We analyzed social networks to identify whether a person had the most influence in the crowd. We analyzed what he indicated and considered it in the requirements identification and prioritization. Some colleagues had already finished their master's degrees; we didn't have PhD. For example, some people had experience, but it was very ad hoc. Today, I see that we would have done much better with the knowledge I already have, including in data science, but the time was also minimal for this management. What we could see was more, for example, a hot topic, you know? It is a more priority requirement that appears a lot. Or a problem that needed to be turned into a requirement. So, there are more techniques for extracting topics, extracting aspects, and extracting social networks to identify if a person is more influential, and then we would take this user where he is. Then, we could manipulate it. Not to manipulate, but to analyze what he indicates to see what to prioritize or bring him closer.

Researcher: Got it. You've already said some things, but the next question is whether you considered the crowd to prioritize these demands in the context of your ecosystem. Interviewee: I'm talking a lot about the previous scenario. I'm going with both scenarios because they help you, right? In the last scenario of developer relations, I see that today, because I have a student researching this, I still see a certain immaturity in identifying the highest priority; we were based on the roadmap. As much as we have access to the product roadmap, we prioritized it, including guiding the relationship with developers and external developers about what is on the roadmap, right? Because it's probably something that we would be charged with and will add to the community as well. Because, probably, new tools and resources will be delivered as this roadmap progresses. Here at [omitted], we don't have a crowd context for the projects we develop, so I don't have a scenario for these current projects.

Researcher: Got it. Interviewee: In the second scenario, we have to prioritize; we initially used feedback tools a lot. This is something that we use a lot. There, we prioritize some of these approaches. What have we been increasing, right? Which is precisely the value perspective. So, we come to discuss what value is. So, is it a value that comes from experience? That's what we consider to prioritize, what's good in terms of experience, what's costly, so then we put it into people, right? Workforce. And what adds monetarily is the perceived financial value, right? So, the most emotional part that makes the person happy is the value that has to do with the money of the project or the people involved. In the budget and the cost part itself, what is the effort to invest in that? So, this has helped a lot of people. Another thing is that we use a lot in this scenario. These are the issue classifications themselves as part of knowledge in open-source repositories. So, cool, you're working on this, man, go there and look at a repository, see how people classified it, if this problem might suddenly be a priority for you at some point, right? However, agile approaches as a tool have helped us a lot from this

perspective. So, we can prioritize, and what we do is also based on the waves, right? If we don't have a vision of the product, let's define a small amount of what we have to prioritize the requirement. So, with today's experience, I know it also fits in with my previous scenario when there were many people. Researcher: Got it. This is very interesting concerning the ecosystem scenario in which it is involved. Now, let's move on to the next question. And in the context of requirements negotiation? Is this crowd considered? Remember that negotiation is also related to conflicts of requirements. Conflicting requirements in the ecosystem must be negotiated with a partner or many partners. Interviewee: This would be the element that, at the beginning, I was very interested in knowing if anything would appear. Because I think the most tense point, which is extremely tense, is negotiation. If you are going to use a smaller project scale, this negotiation phase is already extremely tense. Now, imagine when this will increase in the amount of perceptions and expectations of the project. And then I'll give you a case that actually happened today. But what I like to use is based on some scientific articles and experience. Everything is about establishing contracts in which all parties win. It's not that the person will sign a contract, but there are ways of making the person commit to the agreement. For example, suppose I send the product backlog for that sprint and ask the person to authorize it after a discussion or reprioritize. In that case, they are already negotiating with me and agreeing with that point. So, this is a scenario that I want to bring, in fact, in the example of a small, in quotation marks, contract that we can establish with the person. And then, if I go to app stores, I also bring my other experience. We also have brief contracts there, right? So, when I want to trade, for example, the interface of a store has changed. Yes, the interface and the operating system's shape have changed, which will impact how the applications appear. I don't know if it's a new way of displaying the button. And there? Some stores, right? They put it as an acceptance criterion for that application to continue in the store or to be submitted and accepted. It meets this requirement, so it's a brief contract that you accept to use that infrastructure. I think a very recent example is tracking, right? Applications must show whether the person authorizes tracking a recent case of something that is not a large contract. Still, it is an example of a contract that is established in a negotiation; look, do you want to use my store? Researcher: So, do you seek to define a contract with the parties, a kind of standard for all members of the crowd? Interviewee: That's right, you captured it. So, these approaches I use in projects today, I used them before, but this is the one, this one generates more confidence for me. Just today, I said I would give you a case, for example, for a smaller project involving another university entity. Still, it is a case of a form that is a susceptible item as an artifact. And the engineering team that I coordinated was starting to feel the delivery; there was a delivery day [omitted]. The other side doesn't want to give up, and we will deliver. It's not a problem. But a problem arose that the students raised. Hey, teacher, they are messing with the form all the time. Today, they said no in the discussions, but that's because the form will continually change. You have to be aware of this because just as you are in a scientific environment, so are we, and so I was listening, right? At the time, I was thinking about how to negotiate. Then it generates a tool, man, what I sent them. What we sent them as a backlog. So, I went to look at the backlog; I already know where I will get it. So I went there, and guys, I want to understand now what the sensitive points of this form are. This is what impacts us.

We deliver, but no one said they won't deliver. I explained to them. I think an important point here to negotiate is to create awareness and awareness. And there are tools for that, right? It's about having someone who is a tool, someone who is a posture of trust. So these stores have an image, right? Google, Microsoft, and Apple have an image of trust. The person can still accept your negotiation because they know they will benefit from something of quality. It's the same thing with us. When we are trying to negotiate, this is true for the crowd. If we can influence part of the crowd, that generates that movement, right? We know, via elections, there are scientific articles about this. We don't need to focus on the whole world. Focus on parts. Researcher: Got it. The next question concerns the approaches you use to communicate these requirements during the other activities we have already discussed. Interviewee: We have it. We send it as if it were a survey. This helped a lot with refining between modules, right? For example, there was a module in which they said, professor, the evaluation criteria were not so clear, and it would be important to improve for the next one. We need teachers to be more technical experts and mentors for the journey. So we always have this research at the project's end and return it. So, we put together a communication, thank you and return the main points and say look at this, we will act this way next time and so on and thank you. We communicate this, and when there is a project meeting, we talk, make points, and look. Even though this point was a suggestion sent, you need to participate. So, in the [omitted], it's like that. In others, it is much smaller, right? Why are there more development projects? I'm not going to consider this as a crowd. It's precisely the retrospectives. A retrospective can't be done with a crowd. We often use retrospectives to give feedback and already have an action for the next sprint. It works a lot. I can also tell you that, for example, when I had lectures, I went there to [omitted] give a conference and received a large set of feedback. We also provided a link to a questionnaire where people could send feedback about the talk. It was not just about requirements but also about how the entire process was conducted. Researcher: Right! I think it's pretty straightforward. Now, I'm interested to know if you know or have used automated or semi-automated approaches to handle user feedback from a crowd in the context of your ecosystem. Interviewee: I have had experience with forums. I have been trying to use them at the university, but they don't seem to work as much. I don't know if it's a generational conflict. And on a more level, my relationships as a teacher and researcher. It works well with students, you put it there, they talk, discuss requirements around subject projects, it's fantastic. I use Discord a lot, I used Slack before, with the tool I can see how they are arguing, I can intervene, they tag me if they need something or tag the monitor, but they resolve it, call the other colleague. So, these tools do help. GitHub or GitLab itself has an issues section where you can contribute. There's Gitter, which is now called a Matrix. It also helps in this crowd context by allowing you to select an issue in Gitlab or Github and discuss it with a crowd. These are environments that you can keep observing. I had a student doing research, but I don't know if he will finish his master's degree. It was precisely to identify whether, in these conversations, we identify a moment of tension in discussion environments around resolving issues. For example, the variation of a hostile and neutral sentiment. And as if it were on an electrocardiogram. At that moment, there is a discussion about something very important. That's right; there's a moment of tension, and we could see that. We

identified a tension, there is a lot of exchange of feelings, it means that people are not able to find a path, so someone has to go there and intervene. As it is a crowd, I have no way, if I am a maintainer of a repository, I have no way, wow, I will look at the whole discussion. Sometimes, I think an issue that is a requirement can be super important, but it's not for the people working on it. I have also used software repository mining approaches. For me, it helps a lot and it is the line that I research, it is the line that I have advanced. So, it helps me a lot with this management issue. Researcher: The last question concerns your opinion on the benefit or benefit of crowd-sourced user feedback for requirements management in software ecosystems. Interviewee: It benefits when requirements management can filter and consider this feedback. You know, and then maybe it's coordinated with some statistical strategies, you know? It's a more statistical process approach because, we know, looking at the scenario, not everything is very important, there are aspects of culture that are also very strong, region, geographic dispersion, other elements that you leave there, I congratulate you for investigating this area because there are many variables involved. It's not just about arriving like this, everyone gets this feedback, no man, then you'll see the cultural difference, how important it is for that region. I think requirements management benefits from user feedback when it can filter and weight that feedback through statistical strategies. I consider feedback important as long as it does not direct requirements management. Requirements management should consider it an important volume to be handled with some statistical control that gives confidence to what is coming from the crowd. Researcher: Got it, [omitted]. We have reached the end of the interview. Firstly, I would like to thank you for taking the time to have our conversation and for your contribution to my research. Second, I would like to ask if you have any recommendations for anyone who could also participate in this interview. I'm going to end the recording, and you feel free to check it out later if you prefer.

11 Quotations:

10:16 ¶ 29 in P8 INTERVIEW TRANSCRIPT

We analyzed software repositories to identify and prioritize requirements. We implemented some scripts to identify and prioritize the requirements in these repositories because we could not manage everything daily.

4 Codes:

● _Mechanisms / ● Requirements change identification / ● Requirements prioritization / ● Software repositories

10:17 ¶ 29 in P8 INTERVIEW TRANSCRIPT

We had to create scripts to retrieve data to identify and prioritize requirements. So, we started using data mining methods because we could not manage all comments daily.

4 Codes:

● _Approaches / ● Artificial intelligence techniques / ● Requirements change identification / ● Requirements prioritization

10:18 ¶ 29 in P8 INTERVIEW TRANSCRIPT

We wanted to prioritize the requirements according to what was most important to users. We went to the app store to do so, but we did not have much confidence in whether we were covering what was needed within the ecosystem.

4 Codes:

● _Mechanisms / ● App store / ● Requirements change identification / ● Requirements prioritization

10:19 ¶ 29 in P8 INTERVIEW TRANSCRIPT

We also used Stack Overflow and sorted the issue by reputation to get a better view of this

4 Codes:

● _Mechanisms / ● Forums / ● Requirements change identification / ● Requirements prioritization

10:20 ¶ 35 in P8 INTERVIEW TRANSCRIPT

The analysis was really ad hoc. We implemented some scripts and generated some visualizations. We used the social network analysis tool. We used topic modeling algorithms and word clouds to identify the highest priority requirements. We analyzed social networks to identify whether a person had the most influence in the crowd. We analyzed what he indicated and considered it in the requirements identification and prioritization.

5 Codes:

● _Approaches / ● Artificial intelligence techniques / ● Requirements change identification / ● Requirements prioritization / ● Social network analysis

10:21 ¶ 35 in P8 INTERVIEW TRANSCRIPT

there are more techniques for extracting topics, extracting aspects, and extracting social networks to identify if a person is more influential, and then we would take this user where he is. Then, we could manipulate it.

5 Codes:

● _Approaches / ● Artificial intelligence techniques / ● Requirements change identification / ● Requirements prioritization / ● Social network analysis

10:22 ¶ 47 in P8 INTERVIEW TRANSCRIPT

Everything is about establishing contracts in which all parties win. It's not that the person will sign a contract, but there are ways of making the person commit

to the agreement. For example, suppose I send the product backlog for that sprint and ask the person to authorize it after a discussion or reprioritize. In that case, they are already negotiating with me and agreeing with that point.

3 Codes:

- _Approaches / ● Requirements negotiation / ● WinWin

10:23 ¶ 55 in P8 INTERVIEW TRANSCRIPT

We also provided a link to a questionnaire where people could send feedback about the talk. It was not just about requirements but also about how the entire process was conducted.

3 Codes:

- _Mechanisms / ● Questionnaires / ● Requirements change identification

10:24 ¶ 59 in P8 INTERVIEW TRANSCRIPT

I have had experience with forums. I have been trying to use them at the university, but they don't seem to work as much. I don't know if it's a generational conflict.

3 Codes:

- _Mechanisms / ● Forums / ● Requirements change identification

10:25 ¶ 59 in P8 INTERVIEW TRANSCRIPT

GitHub or GitLab itself has an issues section where you can contribute. There's Gitter, which is now called a Matrix. It also helps in this crowd context by allowing you to select an issue in Gitlab or Github and discuss it with a crowd. These are environments that you can keep observing.

3 Codes:

- _Mechanisms / ● Requirements change identification / ● Software repositories

10:26 ¶ 63 in P8 INTERVIEW TRANSCRIPT

I think requirements management benefits from user feedback when it can filter and weight that feedback through statistical strategies. I consider feedback important as long as it does not direct requirements management. Requirements management should consider it an important volume to be handled with some statistical control that gives confidence to what is coming from the crowd.

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

11 P9 INTERVIEW TRANSCRIPT

Text document, 2 quotations

P9 INTERVIEW TRANSCRIPT Researcher: I have started recording now, and we are going to start the first part of the interview, where I will present some concepts that will be important for us to have the same understanding of the questions that I will ask you in part two of the interview. OK? Interviewee: OK. Researcher: The first concept to come in is software ecosystem because, in general, it is not a well-known concept in practice. However, I have brought a straightforward definition: a software ecosystem is a group of software projects developed and co-evolved in the same environment. When we think about a software ecosystem, we have a technological platform common to several software projects that co-evolve around it. These projects are developed in the same environment, in the environment where this platform reaches a group of actors. Among these actors is the central organization that owns this platform, and the developers internal to this organization help in the development of the platform or other projects. However, external developers have also developed these projects. In addition, the end users of the products offered by the multiple projects are also included. A more practical example that I present about software ecosystems is the Amazon ecosystem, where an e-commerce product has become pretty popular. Still, we have several other products developed around this platform today, such as Prime and AWS. Prime is the streaming service, AWS is the cloud, and others. OK? Interviewee: OK. Researcher: Moving on to requirements management, I am differentiating between requirements development and requirements management in my research. Requirements management is concerned with following the life cycle of a requirement and thinking about the changes that may occur with it, OK? Interviewee: Ok. Researcher: So, which activities are most related to requirements management? The communication of these requirements, change control, negotiation, prioritization, and requirements traceability. The questions I will ask you later are also in this line, focusing on changes in requirements and activities related to user feedback. Interviewee: Understood. Researcher: The last concept is related to user feedback. User feedback is an important resource in modern software development, often containing requirements that help address user concerns and desires for one or more software products. There is an area in requirements engineering that explores the identification and analysis of feedback from a crowd through automated and semi-automated approaches, which is crowd-based requirements engineering, which began to be explored in 2015. Do you know it? Interviewee: I don't know. Researcher: I give an example to explain this; today, we have the popularization of smartphones and the popularization of applications. With that, we have a tool: reviews in app stores, right? The evaluation of these applications in an app store, correct? People can access this functionality and give stars and comments on apps, but what's the problem? You often have thousands, hundreds, and even millions of comments, depending on the application. This information, which can be very diverse, can have relevant information about the requirements, but it can also have irrelevant details, okay? Interviewee: Yes. Researcher: Crowd-based requirements engineering is concerned with creating approaches, whether automated or semi-automated, to collect this information that is available in different repositories, to analyze user feedback, and then assist through concepts like collective creativity or collective intelligence, defining or managing

requirements,? Interviewee: Yes, yes. Researcher: Right, [omitted]. Here, we end part one of the interview. Do you have any questions about the concepts? Did you already know these concepts? Were the concepts clear? Interviewee: I Did not know. I found out about it in the questionnaire you asked me to answer, and it became much more apparent after your explanation. I'm involved in this context and didn't have much idea. Researcher: Great. Now, let's move on to part two. Now, we are focused on what you said. We still don't know how someone at the cutting edge can help with your work as needed. Considering this, someone who is the user is a crowd, talking to previous interviewees involved in the [omitted] ecosystem. I understood that in your scenario, we have two types of crowd: the crowd that would-be users and the citizen client of [omitted] who uses your systems. The user crowd is also composed of agency managers or employees spread throughout the [omitted]. So, we have these two types of crowds, okay? So, considering this, have you ever taught these crowds to identify change requirements? Interviewee: Yes, we consider it minor. As far as I know, when a user encounters an error or a problem in the system that could cause changes, this users open a ticket that is evaluated to see if it really was an error, and then we analyzed this error. Researcher: In this internal evaluation that you say, possibly someone internally first evaluates and checks this error before it reaches you? Interviewee: That's right, and after that, we'll check to see if it's an error. Sometimes there is an error, like, I don't know, the amount credited was wrong. Was it awful? You know, someone will take it and show it to the user. If it's an error, then they send it to us. And then we see if it's wrong. If the problem is with us, we fix it. We are now correcting a priority. Researcher: Speaking of priority, the next question is about that. You said you have a pile of requests. How do you prioritize these demands that come to you about a change in requirements, bugs, and even new requirements? Do you consider the end users of this crowd to prioritize? Interviewee: The manager prioritizes, the manager prioritizes all demands, and then an error also becomes a demand. So these priorities are made by the manager, and he says what is a priority. We can even talk. We can explain that we are in the middle of building demand and that it is a priority. You don't have to do it. Researcher: Then we also come to the next question about negotiation. Negotiation is about conflicting requirements. You may work with managers who have different and inconsistent requirements for the various SECO products. If there is a conflict, how do you negotiate these demands? Do you consider the crowd? For example, what is most requested? Interviewee: No, not in my scenario. I believe that if there were negotiations, it would not have been aimed at the crowd either. Negotiation is more focused on time and cost. You could even be like that and stop thinking about the number of mistakes that will happen. In other words, the number of people suffering from this error. Researcher: The next question is, how do you communicate requirements with this crowd in your ecosystem? How do you receive this feedback from the manager? This question is related to how you would receive feedback from a crowd. How do you usually receive from these people? Interviewee: So, our feedback for our line of work is that when I say a line of work, I say treadmill. In a typical scenario, in the best of all worlds, it is daily, as we present, when a demand ends. Researcher: Do you have a daily conversation with your manager? Interviewee: We have a daily team conversation. The manager participates in our daily meetings. It is a development meeting that serves as an inspection of the team itself. Managers

can participate they can, but this tends to inhibit the times when someone says they have made a mistake. But the review always happens on the last day of the sprint. Ours here for three weeks, 21 days. And yes, it is a meeting in which managers have to participate. We present a formal form, and what has been completed is completed. Then, we receive our feedback. But this way, communication is not stopped at that moment. At any time, I can present advances. I finished a week ago and waited two weeks to show it. No, I can arrive and present because I may have understood something wrong. The manager may have had an idea, changed something, and as soon as I finished, I explained it to him, and so on. Researcher: Cool, you are showing your communication to him. And how does his communication work for you? Interviewee: During the review or planning, in theory, talk to the product owner, and the product owner brings it to us. The conversation with the product owner is open when he needs it. It's not like that, every week there's a meeting, every day there's a meeting. Researcher: What about the tool support to formalize these points and create this flow? Interviewee: We use GID from IBM and Rational. Researcher: So, do you instantiate the result of the meeting using the tool? Interviewee: Yes, yes. Researcher: The next question is whether you know or have used any automated or semi-automated approach to manage requirements coming from a crowd in the context of your ecosystem. Interviewee: I don't know, and I haven't used it. Just the tool I told you about. Researcher: OK, since you didn't use it, the next question is whether you consider that feedback from a crowd benefits your work when managing requirements in the ecosystem. Interviewee: I don't know, why having users' opinions is essential, mainly because they will have ideas, as you said about the other person you interviewed, we will sometimes have an idea of small needs that will make a difference to them. And they will somehow bring about a change, an update to the system that may not even be business-related but actually procedural, and that will make it easier for those using it, right? So, in that sense, I think it's perfect, you know. I think it would be good, yes. It's because there's a lot of internal change, right? As I said, there are things that a manager will use, there are things that half a dozen people will use, and these half a dozen people who will use it, we talk to them once a month, yes. So we have this feedback from these people. They are the ones who open up the demands, so, for example, is it working the way you wanted? It's super great. This open channel for them to bring suggestions is what typically happens, so if you are going to consider the amount of improvements, even if small, that could arise from feedback like this. Considering all users and, in some scenarios, even their customers. I think it would be great to have this feedback from users, but at the same time, I don't see any real gain. Most of the demands we have represent the needs of a few people. Researcher: Right, [omitted]. We have reached the end of the interview, I thank you very much for your time, it turned out to be right within the agreed time and I would like to take this opportunity to ask you for recommendations for more people that I can interview in your context or others. Interviewee: I'll check here and send it to you by email then. Researcher: Ok.

2 Quotations:

11:1 ¶ 38 in P9 INTERVIEW TRANSCRIPT

As far as I know, when a user encounters an error or a problem in the system that could cause changes, this users open a ticket that is evaluated to see if it really was an error, and then we analyzed this error

3 Codes:

- _Mechanisms / ● Issue trackers / ● Requirements change identification

11:2 ¶ 78 in P9 INTERVIEW TRANSCRIPT

I think it would be great to have this feedback from users, but at the same time, I don't see any real gain. Most of the demands we have represent the needs of a few people

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

12 P10 INTERVIEW TRANSCRIPT

Text document, 4 quotations

P10 INTERVIEW TRANSCRIPT Researcher: [omitted], I have just started recording, and now we will start with part one of the interview. In this first part, I will introduce you to some concepts that will help us have the same understanding of the questions I will ask you in part two. All good? Interviewee: Yes, yes. Researcher: The first of these is software ecosystems. I brought up a definition of ecosystem that I consider more straightforward to understand. Lungu defined software ecosystems as groups of developed and co-evolved software projects in the same environment. In general, this environment has a common technological platform and the software projects that surround this platform. Furthermore, a set of multiple actors interact in this environment. Some examples of these actors are the central organization of the ecosystem and the internal developers who are part of this organization. These external developers can develop add-ons or products made available in the ecosystem and the set of users who use these products. Interviewee: Okay. Researcher: An example of an ecosystem I gave in interviews is the Amazon ecosystem. Amazon has become well-known for its e-commerce. However, it also has several other products available on its platform today. Prime is the streaming service, for example. AWS is an interconnected cloud service that evolves together in that environment. Interviewee: Okay. Researcher: The example you may be familiar with is that of [omitted] at [omitted], where the Central organization is [omitted] and [omitted] plays the role of external developer. [omitted], [omitted], and [omitted] are examples of software projects that co-evolve in this environment, right? Interviewee: Okay. Researcher: Another concept I wanted to introduce you to is requirements management. The focus of requirements management is to monitor the changes that occur in the life cycle of a requirement, so it can be related to identifying bugs, improvements, or anything that can influence these changes to requirements. I present the activities related to requirements management here. Change control, version control, traceability, and other communication, negotiation, and requirements

prioritization activities. Interviewee: Okay. Researcher The last concept is a concept you may not know in theory, but you should know it well in practice. It is the concept of crowd-based requirement engineering. What would that be? This is when considering a multitude of users in the requirement activities. Typically, you have a narrow, representative set of users to define requirements. Correct? In your case, you cannot listen to all the students and teachers because many people use the different products in the ecosystem. Crowd-based requirements engineering includes all approaches that engage a crowd of primarily unknown people to perform requirements engineering tasks or provide requirements-relevant information. It is related to user feedback, an important resource in modern software development, often containing requirements that help address user concerns and desires for one or more software products. A crowd in software ecosystems is a large heterogeneous user base giving requirements on multiple channels with multiple stakeholders. Today, we have products that thousands of people use, so we are moving towards a concept of collective intelligence. I don't know if it has already happened, but it could have happened to you because of a [omitted] problem, it could have been noticed by a student in the crowd, or even a solution that was good and you hadn't thought of could still have come from a teacher. Has this ever happened? Interviewee: Yes, it has happened a few times. Researcher: The idea behind crowd-based requirements engineering is to have an automated or semi-automated approach to looking at these comments and finding solutions. Because you can receive hundreds of emails, looking at them individually to understand what is happening is very difficult. Interviewee: It's true. Researcher: Did you understand all the concepts? Are there any questions left? Interviewee: Yes, I understood. Researcher: Now we go to part two, the questions part. So, considering the concepts I presented to you, the first question is whether you have ever considered a crowd to identify or change requirements in the context of your ecosystem. Interviewee: Yes Researcher? How did I manage this? Interviewee: We identified it coming from the users of this crowd, but this is always creating a bridge with day-to-day experience. For example, a significant evolution was made in the system. Discipline blocking and course blocking plus discipline, because for each discipline the student was enrolled in, he or she made numerous requests. I want to lock 5, so there were five requests. There were five student processes: they had to go to course coordination, and they had to go to the protocol. So we identified, in addition to the user himself who suggested this to us, that it be made available in the system itself. We realized that this would have a direct impact on the issue of efficiency of the work carried out within [omitted], so we were able to list and prioritize this demand as a demand that should be met to reduce the paper flow, reduce steps to achieve an objective, which was easily applied within the system. And we did it. Nowadays, [omitted] chooses not to do it like that, but anyway, we have taken many steps backward. We managed to improve many things, but they were falling into disuse. That was really for management reasons, right? Who's on your mind these days? Researcher: And thinking about it, as you considered user feedback from a crowd, what type of approach did you use to be able to identify these requirements? Tools? Or what kind of approach? Interviewee : Our plan was based on some suggestions for improvement that we had already received from the students, the coordinators, and the professors. As we cannot serve everyone simultaneously due to limited staff, we must prioritize, ok? We have made a list of

improvement suggestions available for voting. We wanted to share the responsibility for deciding what was a priority with our users, who are the people who, in fact, use our systems. We made a list available with three suggestions for improvements that we had already received from students for them to list, in their view, which would be the priority. Researcher: I think you just answered the next question. Have you ever considered a crowd to prioritize requirements or requirement changes in your ecosystem? Interviewee: For a long time, we just tried to put out fires until we managed to stabilize the systems. From then on, we started thinking about improvements and then had a goal. Look, in semester X, we need to make three more improvements available to the system: two for students and one for teachers. We tried to manage this kind of internally. We wanted to take away this one-way responsibility a little and share this with our users, who would use the services daily. Researcher: To prioritize, you should consider using the voting technique. And before doing that, how did you prioritize these demands that arrived? Interviewee: We tried to see which improvement would have the most significant impact. If it was a particular thing that occurs once a semester, then we understood that it wouldn't have such a significant impact. It could be placed on the list in a bit of place towards the end. And if it were, I don't know, an improvement in a virtual class that would impact all courses per semester and reach a more significant number of users, then we would prioritize that demand. Researcher: Got it, cool. The next question is, did you consider this crowd to negotiate the requirements? Let's think in two ways: internally with [omitted] and externally with [omitted]. Interviewee: This negotiation always started within the [omitted]. So, this more direct negotiation only happened with two groups of users: coordinators and teachers. This melee existed more in this context, often, as we had a biannual training plan and everything. These suggestions were presented at these moments, and there was even a period, a set aside of a particular time within these training sessions, for them to contribute suggestions. We made this internal assessment of the feasibility of meeting the demand, considering the time we would need to develop that solution and the availability of staff. This has always been a significant factor within [omitted]; there are few developers, which has significantly impacted decisions. And it was done that way. Researcher: Got it. And when did you need to negotiate with [omitted]? How did it work? Interviewee: With [omitted], for example, we only presented the demand because the entire assessment had already taken place previously. This is within the scope of [omitted]. In [omitted], I don't know how it got there, but in the context of [omitted], it got there that way. We already had demand. We have already presented the designed requirements. For example, in a demand for withdrawal, the incoming student cannot withdraw from all subjects. If the student requests a lock in all subjects, the system needs to return, and that is not a subject lock or a course lock. So, we have already arrived halfway with the solution designed for development. Researcher: More of a presentation flow than a negotiation flow. Have you ever participated in any [omitted] negotiations with ecosystem partners? Interviewee: From [omitted], we receive updates and put these updates into production. We merge with what we already have that is different from their code. We don't have much participation in providing solutions, although this is permitted and possible within the network context; I have never seen this happen, even though we have already developed solutions that even [omitted] considers interesting. Like one we created for university

restaurant service, for example, the RU queue management. Researcher: Cool. So, you've already developed some internal things that could be used on the platform. Interviewee: That's right, but we end up using the solutions developed by other partners and by [omitted]. Practically 100% we absorb [omitted] solutions. Researcher: Okay, let's move on to the last questions. How did you or do you usually receive this feedback from this crowd, whether from students, teachers, or other actors in the ecosystem? Interviewee: I think that at this point, in particular, the management of all this could be improved. We didn't have a feedback channel to say whether something was good or bad. So, what we tried to do were tests that focused on that. We gathered different groups of users and asked them to test different sets of features and present some improvement suggestions to us. There wasn't a very defined flow or technique that was a point of improvement. Researcher: I'll give you an example now. As a teacher, I used [omitted] and found some to be a bug, some unexpected behavior, or some suggestions for improvement. How did my feedback reach you? Interviewee: I can tell you very honestly, these requests arrived via WhatsApp, by mailing list, it was very informal. That's why I say there wasn't a technique, there wasn't a flow, there wasn't a very well-defined methodology, some were more judicious, they opened a process in [omitted] sending it there with some suggestions, presenting some problems. This was very common, for example, in a report that did not provide correct information. However, that was the most formality we had. Researcher: So, from what I understand, it could be through a process based on a tool, such as via email or WhatsApp. Interviewee: 95% informal. You could meet me there in the central building, stop to talk. Researcher: It could be in person. The next question is whether you know or have used any automated or semi-automated approach to manage these incoming demands. Whether via WhatsApp, email, or in person. How did you organize this? Interviewee: Internally on my team, I tried to organize all of this on Trello, I used Trello there to try to organize things. What was under evaluation, what was in service, and what was for user testing? I wanted to manage it that way. Researcher: I consider using a tool like Trello, for example, semi-automated, because you need to put the information into it, read the information, and analyze the info yourself manually. And do you know of any automation? Interviewee: Maybe I know, but I couldn't tell you. Maybe you're giving examples, and then, oops, I know that one—maybe Power BI or something like that. No, I didn't use it. Researcher: The last question is whether or not feedback from this crowd benefits requirements management in your ecosystem. Interviewee: For sure. I think user feedback benefits requirements management by reducing decision-making centered on the manager or his experience. The decision centers on the end user, who will use the systems, not the manager, so I think it's essential. Researcher: Thank you, [omitted]. We have reached the end of the interview. Thank you for your time, and I ask that if you have someone to recommend, please let me know.

4 Quotations:

12:1 ¶ 46 in P10 INTERVIEW TRANSCRIPT

We have made a list of improvement suggestions available for voting. We wanted to share the responsibility for deciding what was a priority with our

users, who are the people who, in fact, use our systems. We made a list available with three suggestions for improvements that we had already received from students for them to list, in their view, which would be the priority.

4 Codes:

- _Approaches / ● Requirements negotiation / ● Requirements prioritization / ● Voting

12:2 ¶ 75 in P10 INTERVIEW TRANSCRIPT

We gathered different groups of users and asked them to test different sets of features and present some improvement suggestions to us

3 Codes:

- _Approaches / ● A/B testing / ● Requirements change identification

12:3 ¶ 79 in P10 INTERVIEW TRANSCRIPT

I can tell you very honestly, these requests arrived via WhatsApp, by mailing list, it was very informal

4 Codes:

- _Mechanisms / ● Mailing lists / ● Requirements change identification / ● Social media platforms

12:4 ¶ 95 in P10 INTERVIEW TRANSCRIPT

For sure. I think user feedback benefits requirements management by reducing decision-making centered on the manager or his experience. The decision centers on the end user, who will use the systems, not the manager, so I think it's essential.

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

13 P11 INTERVIEW TRANSCRIPT

Text document, 2 quotations

P11 INTERVIEW TRANSCRIPT Researcher: Okay, we started recording, right? Interviewee: Okay. Researcher: In this first part of the interview, I present some important concepts for the questions in the next part of the interview. I present these concepts also to confirm our shared understanding of them. The first concept is that of software ecosystems. As you know, SECO has several definitions. I am using Lungu's definition, which defines software ecosystems as groups of projects developed and co-evolved in the same environment. I also present some SECO examples, showing a common technological platform and how these software projects evolve around this platform. Furthermore, I highlight that multiple actors are involved. The central organization, external developers, and users. The Amazon

ecosystem, which I imagine you are familiar with, has a platform and multiple products around that platform. Interviewee: Yes, everything is fine! Researcher: the second concept is requirements management, which is the requirements engineering process that has activities related to monitoring the life cycle of a requirement and controlling the changes that occur in it. Activities such as change management, traceability, version control, prioritization, negotiation, and communication of requirements are related to requirements management. Interviewee: Okay. Researcher: The third and final concept is crowd-based requirements engineering. This is an area of requirements engineering that considers user feedback coming from a crowd. It seeks to identify this feedback and analyze it through automatic and semi-automatic approaches to understand the voice of users and use this in the requirements process. User feedback is an important resource in modern software development, often containing requirements that help address user concerns and desires for one or more software products. A crowd in software ecosystems is a large heterogeneous user base giving requirements on multiple channels with multiple stakeholders. An example is the analysis of reviews carried out in application stores. A popular app has hundreds or thousands of comments that reflect the user's views from various perspectives. These comments can help identify changes to requirements, for example. This area of requirements engineering seeks the crowd's collective intelligence to identify critical points for products and can be used to identify important information for the ecosystem. With this concept, I end the first part of the interview and ask if there were any questions or if you have any comments. Interviewee: Everything is fine. Researcher: So, from this initial exposure, we can start part two of the interview if you have ever considered a large group of people that is a crowd to identify or change requirements. Interviewee: No, because things in the ecosystem were already in progress, so that was already closed. It was a closed group; few people were involved, even if external or remote, but there were few people. It was something much more closed. So, I don't think I would even know how to handle the crowd issue. At most, I think I'll do a questionnaire. Researcher: Let's think about your previous context. For example, let's think about [omitted]. His requirements were identified by the vision of the team that was developing, by the vision of the managers, coordinators, by the vision of the department, from what I remember, I think we already talked about this. How did the identification of requirements occur in this ecosystem with many people involved? Interviewee: The department had no influence. The department was just the place where there were teams. The developers were very hands-on, implementing the versions out there. But there was even, in that [omitted] modeling article, a role that was the aggregator, it was not an intermediary, it has an extension of the SSN. This person is closely associated with the manager you mentioned. One specific person, but I had about 3 or 4 who had all the knowledge. They specified in general, for example, that a new institution wants to use the manor, but with particularities. This happened a few times. This person played midfield, traveled, and talked to one person and another. Researcher: Could it be the idea of a partnership manager? Interviewee: Yeah, he wasn't that technical. He wasn't a technician. Because [omitted] had so much technology that he could not do so, he was the business guy. He knew the main features at a high level and sold the idea. He made presentations, advertised, and showed the numbers to those interested. Let's do it, let's go! What do

you need? What is there today that we don't meet, and what do we do to meet it? So he's the business guy; he's the guy who knew the requirements at a high level and was going to pass them on to someone. Then, some more technical people were responsible for the requirements; they were the ones who would translate what the manager brought to the developers, and this specific person was the primary source of requirements. Researcher: Yes, at the same time, it was a source of requirements from several partners. Interviewee: Exactly, to the point that, for example, internally within [omitted], there were three different versions of [omitted] coexisting, all web, plus the mobile version that had different requirements, some were improvements, others were because there was no more like moving around and with external partners, who were something different, it was another focus. So, it was simply just for use. So, there were a lot of requirements; then, there was a requirements management problem. Researcher: So let's think about this scenario, how to manage these requirements that come from different partners that are different. Was everything in the same repository? Was it the same team that worked? Did this manager condense everything? Interviewee: Sure, the manager brought the most business part, but he didn't get his hands dirty. He brought a lot of text, which he understood, and then he passed it on to these others who I said were somewhat more technical with requirements. Those guys broke it down and held meetings with the development team. The teams were very volatile, that was a problem, it always has been. Many interns, scholarship holders, and professors come and go. Then a contractor came, which was very volatile, and they focused a lot on these two's minds. Few developers stayed for a long time and probably still do today. In short, they were hired servants. But like that, I saw documentation of all kinds, documentation of use cases, lighter documentation, already moving a little more towards the agile side. So, it changed a lot. I think the documentation format depends on the time. Researcher: Cool, this scenario is very interesting for the following questions. Considering this boom, the large number of partnerships, and the multiple existing versions, the next question is how the crowd prioritizes the requirements. You said there were several versions and partners, but I understand that the team was the only one, right? Interviewee: That's right, but that core was always the same. I say, at least for about ten years. Researcher: So, how did this group prioritize the requirements, and if it took into account the crowd, if it only took into account the opinion of this business manager, or if there was some approach? How did this prioritization occur? Interviewee: I participated in some meetings, and I remember that the team seemed very cohesive and lucky, right? So it was easy to discuss. It was easy to get right. That doesn't happen with every team. But, generally, when things already came from this business person, they went from a high level, who knew if the partner needed a feature, a report on I don't know what. Then, he wants the ABC field. Was it something like that? The team specified, made proposals, had a purpose and was doing something like that, so it was something more. I can even say that it worked well because some people already had experience with the platform. Now, this decision didn't involve everyone. There were times when there should have been maybe 20 people involved, but today, there might only be five or not even that, so today, it's only the people who know. Researcher: Right, ok. The next question is about requirements negotiation in this context. Sometimes, there is confusion about prioritization and negotiation because they are closely related. But when we are

talking about negotiation, we are focusing on conflicting requirements. So it's when I have a partner A who asks for a change, for example, let's be very simple, obviously what color would not generate this conflict, but like this, partner A asks for a blue color and partner B asks for a green color. You won't be able to achieve both, either blue or green, or you'll choose a neutral color, I don't know, because you've agreed with both of them that it won't be their option. So, this is the negotiation context. There was this context there; how was it carried out? Interviewee: I believe not in this specific case. I will give an example of an interface. There was already the visual identity that was that. Point. All partners used it. In this context, I think there was no discussion on this aspect because they always focused more on functionality. It's about the features. I remember each partner had different branches, so Partner A had Requirement A and Partner B had Requirement B. For example, even though you said one was blue and the other was red, they walked differently. Researcher: Got it. Interviewee: I remember there were some discussions, like, one day, we're going to take all these private things and throw them into a complete version of [omitted]. Then, the time came, and we saw specific things about a partner that were only their own. But if it were an excellent idea that everyone could benefit from, one day there would be an effort to, say, go to the main branch of solar, but generally, when these issues occurred, in this context of negotiation, there would be no problem because everyone worked isolated, so there was no problem. Researcher: And for features in the main branch that required negotiation, was it only done internally with the team and did not involve a partner? Interviewee: It was internally because [omitted] strictly speaking was made to meet [omitted], which are the blended courses at [omitted]. So, the decision of what the platform would be like was up to this team. This team had full pedagogical support. Then, there were external people, people from pedagogy, and people on the usability team. So, everything was resolved internally. Researcher: Considering this scenario of partners, which were many at that time, how did the communication of requirements occur between this business manager and the partners? Interviewee: I remember that he traveled and physically moved around. I remember that sometimes he would go to [omitted]. Sometimes, he went to another location. So, in this shift, it was this time to sell the idea or close the partnerships, bring the main requirements, the new ones, what was needed, and so on. So, this was well before the pandemic, right, so he traveled and it was a different context. Today, I don't think that will happen anymore. He talked to the more technical managers, who are intermediaries, and they gave the general view of feasibility: it works, it doesn't work. We are already behind on versions, deadlines, that sort of thing, but really management. It was so based on these various functionalities at a high level. And then there had to be this step of these people who were closer to the team, and there were a lot of meetings, in person, in a small room, with the developer and the staff so that they would go over these requirements, ways of documenting, specifying these things. So, everything was very in-person, and there were, let's say, these three layers, right? And there? The mechanism was probably email. Maybe we could convene and discuss it like that, but I remember the communication was basically in that sense. In other words, it was a hierarchy from the highest manager to the lowest developer. Researcher: At the tool level, how do you correct this communication beyond face-to-face? Interviewee: The platform has a specific functionality so that users can record their feedback. Then the users would

be professors and students in general. Researcher: That's right, so they only use the environment. How many partners more or less existed in the best phase? Let's think about it like this. Interviewee: I will think about versions. I remember there was an Android mobile version, then IOS. The web version, I clearly saw, had three versions. There was a version for some institutions, but I don't know if it was [omitted]. There were [omitted] people. They had some partnerships, so they used several things [omitted] to train primary education teachers. Some of these are the same version, and some of these are custom versions. The time came when all of this coexisted. Researcher: So, the number of partners was also significant, right? It's not just teachers and students at the forefront, but these organizations, too, correct? Interviewee: If you pass in this context, yes. For example, in [omitted] education, many teachers are in the network. Researcher: The partnership you made with [omitted] as a paid employee? This group representing these partnerships can also be considered a crowd, right? Because no number wasn't just ten people, it wasn't just 5. There should be more, right? Interviewee: Yes. I consider it in that sense—each has different needs and interests. Researcher: And how did this communication with these people occur? Interviewee: Usually through the business manager. Researcher: The next question is whether you know or have used automated or semi-automated approaches to manage this need in the context of partners in the ecosystem we are discussing. Interviewee: No, neither one nor the other. Researcher: The last question is, what is your opinion about considering the feedback of many people for requirements management in the context of a software ecosystem, do you think it is beneficial or does it hinder it. What do you think about listening to the opinion of this number of people instead of what we do today or did in the past, which is to listen to a representative group of stakeholders? For example, you will take five users here, listen, and I will get the idea from that. What do you think about looking at this feedback from around the world? Interviewee: First, I think crowd feedback benefits in quantity, as it can indicate trends and help prioritize. So, this is already a benefit, and, logically, one person's comment may not have any influence, but comments, as you said, if you have 100, 200, or 300, can indicate trends, which can help me prioritize. For example, users complain about this a lot, or they suggest this a lot. For me, this already indicates what should be in the new version. And so, with many people, this can make prioritization easier, bringing it to the more traditional side, right? The benefit is more significant, so it helps management with directions. For example, two or three people say good things about my application, ok? They use it all day, ok, but 500, 600 people are saying it badly. Something doesn't smell right there, right? So, this is a business direction, or whatever, to see if the user target audience is appropriate, I don't know. In this sense, the crowd issue will always be better if there is an automated mechanism so as not to have to read things manually. I initially saw this higher-level part of the business as a benefit, and then the consequences could go down to development. Researcher: Thank you very much, [omitted]. We have reached the end of the interview. I will stop the recording and ask if you have anyone to recommend for future interviews.

2 Quotations:

13:1 ¶ 65 in P11 INTERVIEW TRANSCRIPT

The platform has a specific functionality so that users can record their feedback. Then the users would be professors and students in general.

3 Codes:

- _Mechanisms / ● Built-in user feedback modules / ● Requirements change identification

13:2 ¶ 89, I think crowd feedback benefits in quantity, as it can indicate trends in P11 INTERVIEW TRANSCRIPT

I think crowd feedback benefits in quantity, as it can indicate trends and help prioritize. So, this is already a benefit, and, logically, one person's comment may not have any influence, but comments, as you said, if you have 100, 200, or 300, can indicate trends, which can help me prioritize. For example, users complain about this a lot, or they suggest this a lot. For me, this already indicates what should be in the new version. And so, with many people, this can make prioritization easier, bringing it to the more traditional side, right? The benefit is more significant, so it helps management with directions.

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

14 P12 INTERVIEW TRANSCRIPT

Text document, 6 quotations

P12 INTERVIEW TRANSCRIPT Researcher: Okay, it's already recording. You already know the first concept. So, I'm going to go through it a little faster. I'm using a straightforward definition of software ecosystems, that of Lungu. Lungu defines software ecosystems as groups of projects developed and evolved in the same environment. I have presented an example that shows the common technological platform, the projects around this platform, and some actors that I differentiate between the central organization, external developers, and users. To make it easier to understand, I give the example of the Amazon ecosystem and the example of the [omitted] ecosystem because I am interviewing some people there. Interviewee: Got it. I don't know [omitted], but I imagine it's for the academic. Researcher: Exactly. The second concept is requirement management. Requirements management focuses on monitoring the requirements life cycle and considering its changes. Activities such as traceability, requirements version control, and change management are mentioned in this context. So, I separated some activities that would be of interest in the interview, such as communicating requirements, negotiation, and prioritization. OK? Interviewee: Okay. I understood. Researcher: The last concept is crowd-based requirements engineering. The idea is to use a semi-automated or automated approach to collect information from a large and heterogeneous number of users, a crowd. A crowd in software ecosystems is a large heterogeneous user base giving requirements on multiple channels with multiple stakeholders. Furthermore, user feedback is an important resource in modern software development. The example I have presented is that of mobile device app stores that allow users to send feedback

through stars and comments. Depending on the application's popularity, this feedback can be sent by hundreds, thousands, or millions of people. Based on this feedback, an analysis can be carried out to identify changes, new requirements, and trends for that product. However, doing this analysis manually is complicated. It is costly due to the amount of data and does not often bring you a return. So crowd-based requirements engineering also aims to help in this context by reading and analyzing this feedback automatically or semi-automatically. Interviewee: Okay. Researcher: In part two, I will ask you some questions related to the concepts I presented. All good? Any questions? Interviewee: No. You can ask the questions. Researcher: The first question in part two is whether you have ever considered a large set of users, which we call a crowd, to identify requirements. Interviewee: We took into account the representatives of the business area that is the demander for the product. And then online feedback is more proactive. We have a large group of students and collect more structured feedback. Researcher: Got it. And what approaches do you use to identify these requirements from the crowd? Interviewee: These are online meetings with Miro for us. Right, with the Miro platform and Whiteboard that we always use. These are face-to-face workshops and brainstorming. We put everyone in the room, and we will debate. It is a semi-structured, one-on-one interview, so we collect it with Slack, right? I've also tried it with the form, but it doesn't work well. Staff tend not to respond. So it is more effective, at least for this, we are eye to eye and question the user, so does he do this? Brainstorming techniques may vary, but from a generic point of view, they are face-to-face brainstorming. Researcher: The next question is related to the prioritization of requirements. If you consider this crowd to prioritize requirements, correct? Interviewee: We don't consider it to prioritize. Management prioritizes. It's an idea for the future I found very interesting. Researcher: How do you usually do this prioritization today? Interviewee: See the prioritization; it comes from the quarterly planning, and we get inspiration from the foreigners through Quartz. We define the goals for the quarters as goals, and we divide the quarters into fortnights and sprints. We will evolve, and no, not necessarily. We deliver everything we plan because it's not fixed, right? Do not block planning. Some elements are must-haves. We call it category and fixed data. It has to happen. Others are business expectations that may change, and others appear to be a higher priority. So, the shape comes mainly from the user's perception. These new business opportunities are becoming more important than they are, despite quarterly planning. Or under sizing planning. We realized that there was very little demand. She turned out to be very big. Then we can drive it instead of taking another one. Researcher: And where do you accommodate yourself, or how do you accommodate changes in requirements? Were the bugs you delivered in a previous sprint that needed to be reviewed in this one? Anyway, how do you deal with this? Interviewee: We have a tracking system for recording bugs and demands. It organizes the cards. Cards have tags and swimlanes. We organize through them. These tags and lanes have a category, whether a bug or a delivery demand, and we categorize and group them there. Researcher: Is this system closed to your team or open to ecosystem partners? Interviewee: It is not close to the team. Anyone involved in the project has access. Worldwide. We have internal development and make the results available to more than 1,200 people. They provide registration bugs. We have more specific people for this, and they also report on Slack. They report more on Slack. It's all concentrated in

Slack, so we have each channel for a purpose. WhatsApp was used for some customer service but leads generally carried out the crowd service. So, there is a button on our website that takes the user to a Whatsapp service with the support of a chatbot. This way, it is allocated to our appropriate service queue instead of sending an email with suggestions. We define the flow as one that can be guided automatically. We will also have an email from Baldi. Instead of sending us emails, you will send messages via WhatsApp. Marketing for marketing part. So it doesn't conflict with Slack in terms of communication, it is a direct communication channel with the end user that in the future we can even use WhatsApp itself to carry out this survey with the crowd, understand? Can you define a data source for a set of people who want to participate in this survey? Of course, he will have a signature. Researcher: Right. The next question is related to negotiating requirements or changes that arise. You talked a little about quarterly planning that adapts to what arises, whether because a demand took longer or shorter. Anyway, other things that came up in the business: how does negotiating these requirements work? You are sharing this negotiation. Interviewee: This is something we need to evolve. We have already mapped this, so we need some framework or tool. A process that facilitates management. We have a CFD diagram, right? A cumulative diagram. This shows that the number of items delivered is half our backlog and constantly increasing. So, we have to have an effective way of prioritizing. How do we, in practice, prioritize who shouts the loudest? So, some stakeholders have a greater power of influence, and usually, they capitalize on and push this prioritization, while other prioritizations are more strategic. They belong to the organization as a whole. For example. Exchange of an ERP. There's nothing to do; you must change it, and then it becomes a priority. For negotiation, people use the same requirement identification process, in face-to-face workshops, that's when we integrate them. Researcher: Do these workshops have a lot of people? Interviewee: We've already brought together 15 or 20 people to deal with the issue, but we've already brought together all the financial aspects and the customer service side. Researcher: Ok, moving on to the last questions. How did you get that feedback from the crowd? How does this communication of requirements occur in the software ecosystem considering multiple actors? What kind of channels do you use internally? Interviewee: We generate minutes. I don't know if it is considered in the search term about the crowd. We have internal development. For example, we have a workflow system that maps business processes and creates flows, right? We make it available to a group of 1200 people. This set provides feedback in the Slack channel on the tool's name in question. If 1200 people is considered a crowd concept, we will receive feedback from this group using this tool. Researcher: How do you analyze this feedback? Interviewee: Manual. I'm not saying that all of them, but in the group of 1000 people, people are invited on the channel to ask questions, and there are almost always doubts. About a flow that could be more performative, a button on the wrong side, with the wrong name, and the person doesn't understand. You can ask questions to help. This help is a collection of requirements. Researcher: Right. I think this is very close, in fact, to the point of research. Consider this crowd that is often overlooked because you have a powerful manager. Consider this collective intelligence, even for a few things, to solve problems. Researcher: Have you ever used automated or semi-automated approaches to manage requirements in this context? Interviewee: For

example, we had a very interesting case. We use Google login. Click on the little Google button and authenticate. We hired an XPTO platform, which natively has a username and password, right? she enabled the Google button, and the Google button is at the bottom. So, it is a product that is changed by changing it and omitting the login and password. He would have to take it down and stop working for other customers. So, he chose to leave it open. Every user went there and tried to enter their login and password for their Google account. Researcher: The last question is whether you consider this feedback from a crowd to benefit requirements management in software ecosystems. Interviewee: Yes. It benefits because we can get feedback from a lot of people. I see a lot as you said and I have spoken a lot with an analyst who works with me, for example, he says “there is a button explained there”. And I tell him, you have to see that there are users A, B, and C. Some users previously worked in a system that had to press enter for the cursor to move from one field to another. So, he doesn't know that he has to take the mouse and put it on the other side so that he won't know. So, if we are going to develop and the end user is like this, it could be a project failure, so it is fundamental, I think. Considering the context in which we talked. Researcher: Got it, [omitted]. Thank you very much for the interview. Before closing the interview, I would like to ask if you have anyone you can recommend for me to conduct other interviews?

6 Quotations:

14:1 ¶ 38 in P12 INTERVIEW TRANSCRIPT

We have a tracking system for recording bugs and demands. It organizes the cards. Cards have tags and swimlanes. We organize through them. These tags and lanes have a category, whether a bug or a delivery demand, and we categorize and group them there.

3 Codes:

- _Mechanisms / ● Issue trackers / ● Requirements change identification

14:3 ¶ 42 in P12 INTERVIEW TRANSCRIPT

We have internal development and make the results available to more than 1,200 people. They provide registration bugs. We have more specific people for this, and they also report on Slack. They report more on Slack. It's all concentrated in Slack, so we have each channel for a purpose

3 Codes:

- _Mechanisms / ● Forums / ● Requirements change identification

14:4 ¶ 42 in P12 INTERVIEW TRANSCRIPT

WhatsApp was used for some customer service but leads generally carried out the crowd service.

3 Codes:

- _Mechanisms / ● Requirements change identification / ● Social media platforms

14:5 ¶ 42 in P12 INTERVIEW TRANSCRIPT

there is a button on our website that takes the user to a Whatsapp service with the support of a chatbot. This way, it is allocated to our appropriate service queue instead of sending an email with suggestions.

3 Codes:

- _Approaches / ● Artificial intelligence techniques / ● Requirements change identification

14:6 ¶ 54 in P12 INTERVIEW TRANSCRIPT

We make it available to a group of 1200 people. This set provides feedback in the Slack channel on the tool's name in question. If 1200 people is considered a crowd concept, we will receive feedback from this group using this tool

3 Codes:

- _Mechanisms / ● Forums / ● Requirements change identification

14:7 ¶ 68 in P12 INTERVIEW TRANSCRIPT

Yes. It benefits because we can get feedback from a lot of people.

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

15 P13 INTERVIEW TRANSCRIPT

Text document, 8 quotations

P13 INTERVIEW TRANSCRIPT Researcher: Okay, the recording started. In part one, the idea is to define a software ecosystem for you. Some people involved in software ecosystems do not know the concept. I brought a definition here that I consider very simple of a software ecosystem: groups of software projects that are developed and co-evolved in the same environment. This environment has a common technological platform accessed by different partners from other organizations and users. Around this platform, there are software projects that co-evolve. In this same environment, I also have a set of actors who relate to both the platform and these projects, which would be the central organization, which is the owner of the platform, the internal developers who are part of this organization, the external developers who are from other organizations or who are autonomous developers who can develop projects for the ecosystem around that platform. The users benefit from the entire ecosystem, both from the platform and from these projects. An example I usually give is of Amazon's software ecosystem. As you can see in the image, multiple products co-evolve. Interviewee: Okay. I understand. Researcher: The second concept is requirements management. Requirements management is a process that aims to monitor the requirement life cycle, focusing mainly on changes to that

requirement. When I am referring to requirements management, I am referring to activities related to requirements change management, the traceability of these requirements, version control of requirements, and also other more transversal activities such as requirements communication, negotiation, and prioritization that end up affecting the requirements and can make them change. Interviewee: Yes. Researcher: The last concept is crowd-based requirements engineering. Well, that's a reality nowadays. For example, mobile applications have become popular and have millions of users. Feedback from these users is an important resource in modern software development. CrowdRE includes all approaches that engage a crowd of mostly unknown people to perform required engineering tasks or provide requirements-relevant information. A crowd in software ecosystems is a large heterogeneous user base giving requirements on multiple channels with multiple stakeholders. Researcher: Now let's go to part two, which is part of the questions. The slides explained that the crowd can be made up of users or potential users, users can be internal to their context, and potential users can be external. Have you ever considered a crowd to identify requirements or requirement changes? Interviewee: The questionnaires we applied were an initiative to identify the requirements of this crowd. Then, we prioritized it with the product owner. We don't take everything and put it into production. We received and prioritized the requirements for the implementation issue with the product owner. The technique was the application of a questionnaire, and the tool was an online tool, Google Forms. Researcher: How were the answers to this questionnaire analyzed? Was it manual, too? Interviewee: The analysis was manual, and we didn't get many answers. So, many, I say more than 500, let's put it that way. We read on, and there were many suggestions; we first categorized them. As it is an ecosystem, we first divided it according to the systems, [omitted], [omitted], [omitted]. And so, the first filter was this. Then, the second filter was within [omitted]; what are the modules? We dragged things into the modules to then do an analysis. We used some hierarchical form of prioritization. It was modularizing the feedback. Researcher: The next question is precisely about prioritization. Do you consider the crowd to prioritize whether the requirements came from that crowd or not? How did you do it? Interviewee: No, the product owner defines the prioritization. He may even have the requirement, but he sets the priority. I think a lot of it is due to the practice of what we use, having the owner, and he has the direction. But no, that answer would be no. Researcher: Is there any prioritization technique? Was there a tool you used for this, or was there more? Interviewee: It was through his analysis that we tried; if we had any doubts, we would have a conversation about comparison, which will give us more advantage. It was to ask him a few questions so he could decide and put it in the queue. Sometimes, he wanted to prioritize everything, but no, what is the priority? We had to ask some questions to queue. Don't keep everyone at the same priority level because that's not a priority, right? Researcher: The next question is now related to requirements negotiation. People even have it; it's a lot, and it's very related to prioritization and negotiation. Negotiation is about a conflict of requirements. So, requirement A and requirement B are required. You can only meet one because they are conflicting. Do you consider the crowd to negotiate requirements in the ecosystem? Interviewee: I do not remember if there were conflicting requirements. They were more specific changes because many things came from [omitted]. We took advantage of many things from

them and tried to adapt our process to theirs. As an ecosystem, it would be simpler to adapt rather than restructure the module and everything else. So, in practice, there were small requirements and no conflicts between them. Researcher: Now thinking about you as partners, how did you participate in prioritization or negotiation, or even in identifying requirements for [omitted]. Interviewee: We used [omitted] and [omitted] to inform [omitted] and other members of the team about the requirements, bugs or suggestions for improvements that could be made. But the prioritization process, I don't remember attending any meeting to "look, let's prioritize this here". In meetings, they always said, "Look, this is the priority now". I think they always looked at [omitted]. They identify what is most important to [omitted] and do it. Suddenly, some legislation, for example, a digital diploma, was a demand that skipped everyone's queue. We do it because it's the legislation, and we must comply. We can also inform about defects and improvements in requirements in questionnaires. I think there is a committee there at [omitted] that prioritizes this, but the cooperative does not participate in this. Researcher: And the decisions, based on what you were saying, let's see if I understand. Despite being cooperative, is power centralized in [omitted], or do cooperative members also make decisions together? Interviewee: I think it is very centralized. I don't know if it's because of the size of [omitted], as we are expanding [omitted]. We were more concerned about looking inside than looking outside—more about structuring the processes to be able to start running. And then we didn't have that much participation with them in that sense. Researcher: But is there any dynamic of weekly monthly meetings or workshops, face-to-face meetings, or something like that so that discussion can occur among the cooperative members? Interviewee: Yes, yes, that was. We also used WhatsApp to have moments of conversation between cooperative members and exchange feedback. Yes, there is this practice. Researcher: Did they apply questionnaires or some tool, or was it just that tool you demanded the most? They demanded something from you in this sense, too. Interviewee: Let's go. [Omitted] was a full-time system that was available for us to send our suggestions for improvement. Sometimes, [omitted] sent some questionnaires about using a certain module or some questionnaire about a certain system, and then they shared the results. Feedback received through these questionnaires was used in prioritization. Researcher: Okay. The next question is, how did you receive feedback from the crowd in your context? Through which channels? Researcher: Questionnaire, basically, and the service systems we have, these systems are full-time. They are available for people to make suggestions. It wasn't one place, but the questionnaire was where we had several suggestions for unified requirements. We apply the questionnaire, one of them or via a call, then we will meet with the module manager, right? Who would be the owner, right? For him to prioritize and be able to enter the production conveyor. Researcher: Are there cases you must send to [omitted]? Interviewee: Yes, when it is very complex, something that will drastically impact a business rule of the module. Then, we request the possibility via [omitted]. But it's more complicated because they serve a lot of people. Sometimes, we do it ourselves and prioritize, right? And you do it internally rather than wait for them to post it for everyone, correct? Only if the module or something is more complete. Researcher: The penultimate question is whether you know or have used any automated or semi-automated approach to manage these requirements in the context of your ecosystem.

Interviewee: To manage requirements, we used Redmine, a tool, and then we moved to Trello. We changed tools over time. In a semi-automated way, correct? It's because we intervene. Researcher: Automated, you know or haven't used it, right? Interviewee: No, not to manage requirements. So, he takes it from all this crowd, organizes it, prioritizes it, puts it on the conveyor belt, and delivers it to you. This type is not practical or semi-automated, but it is interesting. It would be interesting if something observed this. What is an ecosystem like? There are several systems and tool modules, and many people access them. It's challenging to make that effort. Researcher: The last question is whether or not you consider that feedback from a crowd benefits requirements management in a software ecosystem. Interviewee: It's a difficult question because, in practice, if we have a lot of feedback, we will have a lot of work managing them manually. Moreover, we will have several views of the same product. For example, what should we follow if the product owner points in one direction and the crowd's feedback points in another? So, does crowd feedback benefit requirements management? I think it depends on the scenario. It can get in the way due to the work that the team will have to do to organize this feedback. However, if there is something that automatically synthesizes it, it can benefit requirements management by improving competitiveness. I consider that the crowd feedback in the context of my organization does not benefit requirements management. Researcher: That was the last question. Thank you for agreeing to participate in the interview, and please let me know if you have any suggestions for an interviewee. OK?

8 Quotations:

15:1 ¶ 16 in P13 INTERVIEW TRANSCRIPT

The questionnaires we applied were an initiative to identify the requirements of this crowd.

3 Codes:

● _Mechanisms / ● Questionnaires / ● Requirements change identification

15:2 ¶ 36 in P13 INTERVIEW TRANSCRIPT

We used [omitted] and [omitted] to inform [omitted] and other members of the team about the requirements, bugs or suggestions for improvements that could be made.

4 Codes:

● _Mechanisms / ● Forums / ● Requirements change identification / ● User feedback tools

15:4 ¶ 36 in P13 INTERVIEW TRANSCRIPT

We can also inform about defects and improvements in requirements in questionnaires

3 Codes:

● _Mechanisms / ● Questionnaires / ● Requirements change identification

15:5 ¶ 44 in P13 INTERVIEW TRANSCRIPT

We also used WhatsApp to have moments of conversation between cooperative members and exchange feedback

3 Codes:

● _Mechanisms / ● Requirements change identification / ● Social media platforms

15:6 ¶ 48 in P13 INTERVIEW TRANSCRIPT

[Omitted] was a full-time system that was available for us to send our suggestions for improvement

4 Codes:

● _Mechanisms / ● Requirements change identification / ● Requirements prioritization / ● User feedback tools

15:7 ¶ 48 in P13 INTERVIEW TRANSCRIPT

Sometimes, [omitted] sent some questionnaires about using a certain module or some questionnaire about a certain system, and then they shared the results. Feedback received through these questionnaires was used in prioritization

4 Codes:

● _Mechanisms / ● Questionnaires / ● Requirements change identification / ● Requirements prioritization

15:8 ¶ 52 in P13 INTERVIEW TRANSCRIPT

Questionnaire, basically, and the service systems we have, these systems are full-time. They are available for people to make suggestions. It wasn't one place, but the questionnaire was where we had several suggestions for unified requirements. We apply the questionnaire, one of them or via a call, then we will meet with the module manager, right? Who would be the owner, right? For him to prioritize and be able to enter the production conveyor

4 Codes:

● _Mechanisms / ● Questionnaires / ● Requirements change identification / ● User feedback tools

15:10 ¶ 68 in P13 INTERVIEW TRANSCRIPT

It's a difficult question because, in practice, if we have a lot of feedback, we will have a lot of work managing them manually. Moreover, we will have several views of the same product. For example, what should we follow if the product owner points in one direction and the crowd's feedback points in another? So, does crowd feedback benefit requirements management? I think it depends on

the scenario. It can get in the way due to the work that the team will have to do to organize this feedback. However, if there is something that automatically synthesizes it, it can benefit requirements management by improving competitiveness. I consider that the crowd feedback in the context of my organization does not benefit requirements management.

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

16 P14 INTERVIEW TRANSCRIPT

Text document, 6 quotations

P14 INTERVIEW TRANSCRIPT Researcher: Okay, the recording has started, right? In part one, I will present some important concepts for the questions I will ask you in part two of the interview. The first concept is of software ecosystems, groups of software projects developed and co-evolved in the same environment. So, when we think about a software ecosystem, the idea is to have a common technological platform with groups of projects co-evolving around it. Multiple actors participate in this ecosystem, such as a central organization that owns this platform, internal developers who are part of this central organization, and external developers who develop these projects around the platform—and finally, the end users who benefit from the products generated in this environment. An example of a software ecosystem is Amazon, which has several products co-evolving around its platform. Interviewee: Right. Researcher: The following concept is requirements management. Requirements management is the requirements engineering process that follows the life cycle of a requirement, focusing on the transformations that this requirement may have. So, in general, it will be related to activities such as change management, traceability, and version control of requirements. So, on the next slide, I mentioned some activities directly related to requirements management. How are these requirements communicated, how are changes controlled, and how are these requirements negotiated and prioritized? So, in the question phase, we will be involved in some of these activities for a little while, okay? Interviewee: Yes. Researcher: The last concept is crowd-based requirements engineering. This concept is related to user feedback, an essential resource in modern software development, often containing requirements that help address user concerns and desires for one or more software products. The idea of crowd-based requirements engineering is to use automated or semi-automated approaches to collect a large amount of information and evaluate user feedback so that this feedback can assist in requirement activities. Researcher: In part two, I will ask you some questions about the concepts I presented. The first question is, have you ever considered a crowd to identify requirements or requirement changes in the context of your ecosystem? Interviewee: Yes Researcher: How was it? And how was this crowd considered? Interviewee: It's difficult, right? I've had huge crowds and smaller crowds. So, of course, the difficulty level will vary the wider your crowd, right? There was a time when the crowd was made up of all the deans and pro-rectors of all the Institutes of [omitted] who met with us in [omitted]. There were several sections for submitting, validating, and prioritizing

requirements until reaching a consensus. Because many of these people have difficulty understanding how it works in a computerized, automated way, they have another view: they were people who weren't even that close to technology. So, it was arduous. At the time, we didn't use the resources we use today, such as text mining, document analysis with natural language, prototype analysis, and interface with image analysis. We didn't have any of these resources at the time. So, it was more traditional requirements engineering. What helped us were the tools that we developed ourselves. To manage the requirement itself, automatically transform it into a task and, from there, monitor these requirements. So, I think that a facilitator for interacting with this crowd was the tool we developed to obtain this feedback.. It's much calmer with smaller groups, although different people are industry managers, each with their own thoughts and vision. However, being a smaller group with the appropriate tools and regular meetings, things don't get out of control. With each delivery, you can evolve the requirements and evaluate them later, whether satisfactory or not. What I think also helps a lot is working with the most agile methodology possible. In my opinion, today, I can only work with Scrum. I can no longer use those traditional cycles that we used 10 or 15 years ago because that doesn't work; at least for me, as a project manager, it doesn't work. So, all the projects we have worked on now in the last, I don't know, ten years, eight years, have always been using Scrum, Kanban, and all of this within our tool that we developed for this. Researcher: Is the tool you developed suitable for your scenario? Interviewee: It is a very hybrid tool, so it lets you use Scrum, it enables you to use PM Canva, it lets you use Kanban, it allows you to use PMbok concepts, but you can use everything with hybridism. So, the manager has a facility, understand? So, I'll tell you what I use today for the tools Scrum, kanban, and Canvas. And I still sometimes do an EAP when the project is more complex, the EAP helps me with the tasks linked there, but all that heavy documentation doesn't. On the software side, I only list requirements within the tool itself, functional, non-functional, business rules, and the class diagram. The class diagram is not in the tool, but I can upload it. Researcher: What other types of methods or techniques do you use? Interviewee: So, it's more recent. We have even used it in the initial demand, we didn't use it in the industry, we used it to fulfill a dissertation to see how things would work, right? So, we are integrating this entire textual analysis part into this same tool. So, the last one we did now was for a dissertation that wanted to implement automation in the legal office. Then we worked with natural language analysis on the PDF documents, and he already had to extract requirements from there. We also used some of it to understand his business and business modeling in BPMN. Then, we developed a plugin that tracked and transformed that into a requirement when necessary. I already identified whether it was a system requirement or a user requirement. Suppose the process was going to be manual. This made meeting the client's requirement easier, which was the people involved in the office because it was a dissertation case study, which he wanted. So it worked. It worked well; it gave a satisfactory result in the end. Now, we are applying it with two more dissertation possibilities, doing all the requirements gathering in this way, even on top of the [omitted] system, which is also there at [omitted]. What makes it easier is not pulling sardines into my tool, but it's because we have everything in one environment. The name is [omitted], now integrated management 4.0. So, everything we need to be in the tool, let's suppose I

want to know what happened in a project I developed in 2016 for [omitted]. I go to the tool, open it, and find out everything that happened. It's all documented for me, so it's much easier, right? Researcher: The next question is, do you consider the crowd to prioritize requirements or requirement changes in the context of the ecosystem we are discussing? Interviewee: So, in terms of prioritization, we've already tried different things, from straightforward things to more sophisticated things, because, like, I'm from the academic field, although there are government projects with industry, I'm an academic, right?? So, I always relate my students' research to what I will offer. So, we have already used multi-criteria methods. Lately, we have been using AHP more. Now, I'm going to experiment with Macbeth, which is another multi-criteria method for prioritizing. Researcher: But then do you prioritize with your internal team, or do you also consider this crowd to prioritize, or do you have a representative, for example, in the case of the deans you mentioned? Did you consider this crowd? Interviewee: No, when it's a crowd, when we don't have a single client, it's always with the client. The internal team can have an opinion on what goes on their Scrum list. What will go into those sprints? However, the actual prioritization of requirements is always up to the crowd. Generally, it happened in meetings and often didn't end at the meeting, right? It was difficult to close, and then it went online, or the method for them to prioritize, the questionnaires for them to prioritize, you know? Depending on the size and complexity of the crowd, some people find it difficult to understand what it means to give weight to the multi-criteria, and then you cannot work with that person in that way. Researcher: How did requirements negotiation work considering this crowd in the ecosystem? Interviewee: Look, in the bigger crowd, so to speak, right? That was the issue with the deans. I don't remember now that there was a conflict between requirements. Generally, a consensus was reached because, although the institutions are autarchies, they function similarly. Now, in the industry, what has already happened about the conflict wasn't that one person wanted a requirement and the other didn't. The requirements conflicts were about the time it would take and the cost it would have. To negotiate these requirements, we considered these factors and presented them to the partners -- for example, time and cost that will be required, and we resolved conflicts this way. Did you understand? Sometimes, one wanted a requirement, and the other wanted that requirement to be a little more complex, but to do so, the project would have to be extended and cost more, and then the conflict resolution was always like this. Look, this is the time and the cost, which was resolved based on that. Do you understand? It was in consensus. Then, the person who defined it was the industry association president. Someone had to define it. Researcher: The next question is, how did requirements communicate with this crowd? What communication channels were used? Interviewee: Generally, it was the tool itself. Researcher: So, deans and people in the industry had access to the tool and could monitor the sending there. As you said. Through the request, right? Interviewee: So, in the industry, they had a contract, especially when it was [omitted], you have to have a contract. Initially, the requirements are all in your contract, and you have to make an addendum with each change in requirements. This is a pattern used for this type of project. In the other project, we generally delivered. In addition to having the tool, as many people did not have access to these things, they had a particular aversion to using technology. The IT people saw the requirements of the tool.

Generally, the manager, the dean received a technical report with the requirements and prioritization, and everything was correct. Then, when the change happened, we sent the change to that report. It all depends on the type of client the person has, right? Some people are completely technological. Some people are not. Researcher: The next question is whether you know or have used automated or semi-automated approaches to identify requirements in software ecosystems. Interviewee: So, this is a part we are exploring now. I think I supervise two or three dissertations in this sense. There's a guy who works in the company's own IT department [omitted], we have a system there to receive requests. And it's always someone, a group of people, who will analyze what will and won't work and give feedback there. Now, we do everything via chatbot and text analysis, with minimal team interaction, so the team is more accessible to get their hands dirty. To develop or service something you can't do automatically. Researcher: And the last question is whether you consider taking this feedback from the crowd into account. Does this benefit the overall requirements activities of requirements management, and if so? Finally benefit, it gets in the way because. Interviewee: We who work with software cannot say that we will not listen to our users or that this is not beneficial. However, I think it's a question of balance between listening to the crowd and considering key stakeholders. We can do text mining and screen analysis, but some things are in the client's tacit knowledge, and we must talk to them. I think these automated techniques will reduce the volume of meetings and the tensions that occur in them because by analyzing the results of the automated approaches, you go into meetings with more knowledge about that domain. I've seen a lot today and even talked a lot with these people. It's the startup people. Startup people don't listen to anyone. They want to do something very quickly and put it on the market. If it doesn't work, you'll fix it along with it, right? And until now, I have argued quotes. With a student who has a startup, he said, no, but the bodies, which I won't say here, are encouraging you not to do anything more in software engineering. That you develop and launch. I said well, there is a severe problem here. Because if you develop and launch and you launch crap, you'll never go back to the market. It's a considerable risk you're going to take. Not everything you'll put on the market just because you did it quickly is good. So, today, what I see in these children, in these young people from startups, is a mind going in the opposite direction. I think everything is a balance. It will never be traditional engineering; we passed that on long ago. I'm not a fan of conventional software engineering, not a fan at all, but I'm also not a fan of crazy, reckless disruption. So, I think people are very much at the edges; sometimes, the academy at one end is still very focused on the traditional, and the market at the other end is going wild toward something that may not work and often doesn't, right? And I also see it happen like this: sometimes, the guy sells a business, sells millions of I don't know what, and behind it is a scam, it a scam, that doesn't work for, at most, a few months because when it bombs user, it will crash. So, this thing is happening, you understand? People who aren't from computing and study algorithms there want to develop it and go to market; it's wild. I don't know how to explain this movement well, and it has the support of some bodies to work this way. Researcher: Do you believe these approaches replace more traditional requirements engineering techniques? Interviewee: Yeah, I think it's a hybrid thing, right? It's a hybrid thing because a lot of what we need to know about the requirement is in someone's head. And then, you still don't have how to put a chip

there to collect information. So, no matter how much you mine the documents or analyze the screen, many things are in someone's tacit knowledge, and you have to talk about them. I think these techniques will reduce the volume of these meetings and the tensions that occur in these meetings. Because when you analyze something automatically, you can go to these meetings with knowledge of the facts about that larger domain. Huh? Even for you to talk to the user, because if you don't know what he wants, his company, how it works, and he also doesn't understand the technical language very well, communication is complex until you reach a point favorable to both of you. Right?? So, I see it as a hybrid method that minimizes this type of interaction but does not replace it. It's that thing; we haven't reached the point, and I don't want it to come to an AI replacing our human interaction, right? But could it happen? I even think that a trend is coming, which scares me because I witnessed a conversation between my student and ChatGPT, where ChatGPT spoke to him like this, I am thrilled, I feel glad that you are satisfied. So he's already trying to simulate the feeling. The degree of interaction you have with ChatGPT is something I can't do. I ask two things, and it's wrong, and I get upset and leave, but this student of mine loves it, so he has a deep interaction, reaching the point of ChatGPT trying to express a feeling for him of Joy. Researcher: Well, [omitted]. We have reached the end of the interview. Thank you very much for your time and conversation. It was very interesting and very motivating too. I'll send you an email later to ask you for more people to recommend for the interview, ok?

6 Quotations:

16:1 ¶ 20 in P14 INTERVIEW TRANSCRIPT

I think that a facilitator for interacting with this crowd was the tool we developed to obtain this feedback.. It's much calmer with smaller groups, although different people are industry managers, each with their own thoughts and vision

4 Codes:

● _Mechanisms / ● Requirements change identification / ● Requirements negotiation / ● User feedback tools

16:2 ¶ 27 in P14 INTERVIEW TRANSCRIPT

everything we need to be in the tool, let's suppose I want to know what happened in a project I developed in 2016 for [omitted]

3 Codes:

● _Mechanisms / ● Requirements change identification / ● User feedback tools

16:3 ¶ 31 in P14 INTERVIEW TRANSCRIPT

we have already used multi-criteria methods. Lately, we have been using AHP more. Now, I'm going to experiment with Macbeth, which is another multi-criteria method for prioritizing.

3 Codes:

- _Approaches / ● Multi-criteria decision-making methods / ● Requirements prioritization

16:4 ¶ 54 in P14 INTERVIEW TRANSCRIPT

we do everything via chatbot and text analysis, with minimal team interaction, so the team is more accessible to get their hands dirty

3 Codes:

- _Approaches / ● Artificial intelligence techniques / ● Requirements change identification

16:5 ¶ 58 in P14 INTERVIEW TRANSCRIPT

We who work with software cannot say that we will not listen to our users or that this is not beneficial. However, I think it's a question of balance between listening to the crowd and considering key stakeholders. We can do text mining and screen analysis, but some things are in the client's tacit knowledge, and we must talk to them. I think these automated techniques will reduce the volume of meetings and the tensions that occur in them because by analyzing the results of the automated approaches, you go into meetings with more knowledge about that domain.

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

16:7 ¶ 40 in P14 INTERVIEW TRANSCRIPT

The requirements conflicts were about the time it would take and the cost it would have. To negotiate these requirements, we considered these factors and presented them to the partners -- for example, time and cost that will be required, and we resolved conflicts this way.

3 Codes:

- _Approaches / ● Multi-criteria decision-making methods / ● Requirements negotiation

17 P15 INTERVIEW TRANSCRIPT

Text document, 10 quotations

P15 INTERVIEW TRANSCRIPT Researcher: I started recording. As I told you, the first part of the interview aims to introduce you to some important concepts for the questions I will ask soon and also help us check if we have the same definitions. Okay? Interviewee: Alright, let's go. Researcher: The first concept is the software ecosystem. SECO has almost a dozen concepts. The concept I am bringing is what I consider to be the simplest to understand in our daily lives. A software ecosystem comprises groups of software projects developed and evolving in the same environment. In general, we have a common technological platform in a software ecosystem and several software projects around that platform. These projects can be

developed by external developers or the organization controlling that platform. Products in an ecosystem can be independent, however, they have some integration link with that platform. In an ecosystem, we have a set of actors made up of a central organization that owns this platform. These internal developers are part of this central organization, external developers, and users of the multiple products of a software ecosystem. An example of a software ecosystem I give is Amazon. I don't know if you know, but Amazon has several products that have been developed and evolved around its platform, and they have some integration. Interviewee: Yes, I know. Researcher: The second concept is requirements management. Requirements management aims to monitor and control the requirement throughout its life cycle, focusing on the changes that may occur in that requirement. So, from identifying a change, how it affects the requirement, and how that requirement transforms. Therefore, requirements management relates to change management, traceability, version control, communication, prioritization, and requirements negotiation. Interviewee: Got it. Researcher: The third and final concept is a crowd-based requirements engineering concept. What would that be? This is when we use automated or semi-automated approaches to analyze information from a large and heterogeneous set of people, that is, a crowd. A crowd in software ecosystems is a sizeable heterogeneous user base giving requirements on multiple channels with multiple stakeholders. User feedback is an important resource in modern software development, often containing requirements that help address user concerns and desires for one or more software products. In short, crowd-based requirements engineering includes all approaches that engage a crowd of mostly unknown people to perform requirements engineering tasks or provide requirements-relevant information. Interviewee: Okay. Researcher: I always give the example of using applications for mobile devices. Today, millions of people use these applications. So, app stores implemented feedback mechanisms to reach this crowd. Reviews in app stores can be done through stars and comments. In those comments, there may be compliments and complaints that can impact changes in the requirements of that product. However, there are thousands or millions of comments on a popular app. Analyzing these comments manually is very costly. However, those comments can also demonstrate trends for that product or the ecosystem. Okay? Interviewee: Okay, even when you explained this crowd-based requirement engineering, I already thought about the evaluation we do in applications. Before you even explained it, I was already thinking about it. It's a way for you to collect in bulk, right? Researcher: That's precisely cool. So, let's move on to the second part of the interview, the questions part. Do you have any questions about this first part? Interviewee: From your explanation, everything was obvious. Researcher: Thank you very much; let's move on to part two. The first question in this part is whether you consider many users to identify or change requirements. Interviewee: I'm trying to think here within our work process, although many people use our systems. I don't know how many thousand students we have here at [omitted], for example, 30,000 or 40,000; I have no idea or even more. And then, you add 6000 servers. Please wait a minute, let me reread it; I'm trying to think. I remembered, yes, yes, we already considered it, do you know when we do this? When launching a new system, we need to collect user feedback. Researcher: How does that work? Interviewee: We do this a lot to get feedback, mainly in terms of the systems' usability and understanding. So, this is

done a lot with the design team responsible for the prototyping part, right? New systems, mainly. But we do, we do. We did it through questionnaires to identify problems and collect any new requirements. We even did this at the end of last year with the [omitted] system. With the group of course coordinators, the management, and some employees, we could identify the possible difficulties they would have in using the tool as it was a new system that we launched recently. We launched it at [omitted], and we launched a questionnaire to collect feedback from these people. On top of that, we could understand whether that requirement was well done or not. So, I didn't remember at the time, but when you said it, the start came, so we did it.

Researcher: And with cooperative members, for example, there is a new functionality. In addition to your vote from your members, they may have ideas you can bring together with them or launch a questionnaire for them so that they can have their opinions. It happens?

Interviewee: We already did the collection, I don't remember, but what we do with several projects, for example, is to collect requirements and ideas from their feedback and do research. The sector is launching a postgraduate course, and we conducted a survey. The management carried out a study to understand the best area to do this postgraduate course that would be of interest together with STI so that we could improve the part implementation in terms of their service, understand? Because of that, we have a lot of cooperative institutions. Sometimes, the institution does not have an IT team that can provide the necessary support. So, we need a DBA when we say that we need an analyst who develops, understands infrastructure, and works with it. Many institutions sometimes do not have it. So their implementation process is prolonged; we even did a survey now to find out the partners who would be interested in doing this postgraduate course, the areas they would be interested in, to be able to close a postgraduate program within STI that understood the partners, right, to be able to carry out a type of training directed to the partners, focusing on the implementation part of the system.

Researcher: I will now move on to prioritization. How do you prioritize? You talked a little about voting. Do you use the crowd to prioritize?

Interviewee: For example, all have access to [omitted], where they can search for ideas or register suggestions. If it's a suggestion we think is viable, we can accept it and leave it open for voting, and then people will vote. Do you understand? So, the suggestions that receive the most votes are chosen. Here at [omitted] we have two ways of prioritizing, we have prioritization via committee, university committee and there is a cooperative committee. The [omitted] committee prioritizes internal demands, and the cooperative committee prioritizes demands from cooperation. So, if we have 30 members, we have 30 members demanding, there is a committee that will look at everything that is coming and what is most important that we need to meet.

Researcher: Are these analyses always manual? So if a hundred requests come in, do they analyze them individually to prioritize?

Interviewee: That, that, many times, it already more or less says what it is and what it needs. Internally, we already know the demand, talked a little with the user, and put some observations in the spreadsheet saying that if it is an impactful demand, what will it involve? We do a rapid analysis to understand the demand size more or less. We don't go deeper because we only go more profound when the committee says we must. Then, we start to delve deeper. The other way we must prioritize is day-to-day demands because we receive demands for adjustments, error correction, audits, data extraction, and data updating.

We accept these five types of demands. The only thing that doesn't have priority in the service queue is the adjustment because it is always like this: you can't do it in a hurry. You have an error in the system where you cannot close a class or enter a note, so it is a blocking error, so we categorize it as urgent. We have it in our Redmine; we can categorize it, say what is urgent, high, medium, and low; any demand and adjustment demands never have a high priority unless it is essential, and when it goes through the leader and the leader prioritizes and puts it in the queue. However, for the adjustments in general within [omitted], we have the so-called prioritization queue, in which each sector prioritizes what is most important. We prioritize; some sectors are looser, and then we don't have that much demand. So, when it arrives, we already categorize it, and since they don't ask for much, they are smaller sectors, so when it arrives, we fit it into this little girl. So, we don't leave. Whoever shouts first doesn't. We need to understand. Please tell me what is most important, and we will take care of it. So, in our internal development process, for example, we are still developing the [omitted], which is the [omitted] system. So, when [omitted] started to be developed, everything was necessary, and we started building the system. The system is live and continues to evolve because we understand that improvements are required. So the technical advisory sector of [omitted], below the pro-rector, sits down and prioritizes. So they say, now in July, I need this demand package. OK! What do you need for August? What is most important for August? I need this thing. So, we will prioritize and analyze, develop, and launch screens for development. So it's very much like that; the basis is that we check with the user what their priority is, and we respond according to our development flow. Researcher: How do you manage after you have prioritized from this committee which external members of the cooperative and your internal ones? Do you weigh external and internal? Do you put it in the same backlog, and are different teams doing it? Interviewee: No, no. Depending on the demand, we have a backlog specific to that demand and analyze how long it takes us to develop. Generally, we put together a roadmap; in the next semester, we will cover A, B, and C. Internal and external. Now, let's prioritize one from the cooperative and one from [omitted]. We make more or less a plan for the semester, for the quarter, for the year, because everything that goes everything that arrives is urgent, everything that comes is essential. Researcher: As time is a little busy today, I will move on to the next question. In the ecosystem, there may be requirements conflicts between the cooperative members or between you and the cooperative members. These conflicts require negotiation of requirements. How does the negotiation of requirements occur in the ecosystem? Interviewee: I don't remember there being much conflict in requirements, but generally, we analyze the impact of what they are asking for within the institution. We analyze whether it makes sense for the institution to respond. If you feel it, we'll answer. If it doesn't make sense, the member is free to develop. Because it needs to make sense. If it doesn't make sense, if it doesn't add value, we won't meet the demand. Researcher: For example, has there ever been a case where you responded to a member's demand, and the other members thought it was terrible? Interviewee: Not that I remember. When we develop, everyone will benefit at the end of the day. Did you understand? That's why we do an impact analysis to determine if it makes sense. Researcher: And a cooperative member may not use it, right? Interviewee: She may not use it, but it's there. Do you understand? So, that's why we don't accept everything that comes our

way; we need to analyze it, and sometimes it's a suggestion that doesn't make sense and can break something else. Researcher: Exactly. Moving on to the final part, I have some questions. How does requirements communication occur in your ecosystem? Through which channels can user feedback reach you? Interviewee: The [omitted] is a forum for doubts. It is a communication channel where they ask questions, and we respond. We have the [omitted] itself. [Omitted] is the systems activity manager. It is our customized Redmine. Anyone who partners with us can access [omitted] to open a ticket. We also have the cooperation service questionnaire, which is the call [omitted], [omitted], which provides targeted service that they do directly with the technical consultants. Oh, I have an error here to upload the environment, what do I do? So he opens a ticket at [omitted], and the consultant gets in touch with him and provides support. There is an error that they report within [omitted] so that we can correct it. Look, I noticed a mistake here in this version. We will analyze the error. If an error occurs in our environment, we forward it to development. If it doesn't happen, the cooperation analyst will explore why it's happening for him and not us. Do you understand? So, we have these channels to communicate, prioritize, and negotiate, so [omitted] is only for cooperative members, and [omitted] is for servers and cooperative members. [Omitted] is a system for internal employees and cooperative members to send suggestions for improvement. Researcher: I understand, from what I know, that these channels are closed apart from [omitted]. Only those who are cooperative members have access, right? Are there open channels? Did anyone who is suddenly send something? Interviewee: On our website, there is contact us. It's already happened, and we receive a lot. We have already received suggestions on Instagram. We have a communications team that is responsible for obtaining all these demands and directing them. Researcher: And what comes through these more open channels? Exciting things are coming, too. Interviewee: In fact, there are a lot of doubts, and people are asking how they can do research. Just now, a request arrived from someone who wanted prints of [omitted] to do some work, but generally, there are more questions than suggestions. Researcher: Now, moving on to the penultimate, have you ever used any more automated tools, for example, natural language processing, text analysis, or something that analyzes this large amount of data to analyze crowd feedback? Interviewee: No, not as far as I remember. Researcher: Ok. In general, you use more semi-automated tools than the ones you mentioned to manage, divide tasks, and analyze, but nothing is more automated. Interviewee: Not for the requirements part. Researcher: Okay, the last question is whether you consider crowd feedback to benefit or hinder requirement activities in the context of your ecosystem. Interviewee: I think it's beneficial, but you need to know how to separate and filter this feedback because we receive suggestions for improvement, bug reports, and complaints daily without asking. If you can identify and prioritize this feedback, it can bring good results. So, the person comes up with the idea, but we don't know about it, and it's so obvious, right? And so, we've already taken advantage of it, we've already made a lot of improvements to the suggestions that come our way, so, I don't think it's a bad thing. I would find it wrong if we set out to meet everything that arrives because we would never be able to do it. So, when we open [omitted] to receive suggestions, what we will do is not explicit. This is not written there. It is a suggestion. So, if it's a suggestion, it's not saying we will do it. I think it is beneficial. I don't see it being wrong. Researcher:

For example, something that was a good idea but ended up being lost there for a long time. Precisely because this analysis is manual, and there could have been human error or even something that was very requested. Have you ever noticed anything like this? Interviewee: Not that I remember, but I've seen something in [omitted] where the person made a suggestion and we didn't have any interaction to respond, but what the person wanted wasn't there in [omitted], it was somewhere else sector and the person was left there in limbo. When I went to look, I saw a possible suggestion there for a few days, a few weeks, and I was distraught because no one saw it. I went there and responded to the user, so it was our mistake to pass over and not respond. By bad luck, the person had a problem and needed to solve it. He played on [omitted], waiting for a response, and there was no response, so I responded and apologized. I guided as I could only do. Researcher: Thank you very much for your time, our interview is over, I will end the recording, but I also wanted to check with you if you have anyone to recommend.

10 Quotations:

17:1 ¶ 34 in P15 INTERVIEW TRANSCRIPT

We did it through questionnaires to identify problems and collect any new requirements. We even did this at the end of last year with the [omitted] system.

3 Codes:

- _Mechanisms / ● Questionnaires / ● Requirements change identification

17:2 ¶ 66 in P15 INTERVIEW TRANSCRIPT

The [omitted] is a forum for doubts. It is a communication channel where they ask questions, and we respond.

3 Codes:

- _Mechanisms / ● Forums / ● Requirements change identification

17:3 ¶ 66 in P15 INTERVIEW TRANSCRIPT

We have have the [omitted] itself. [Omitted] is the systems activity manager. It is our customized Redmine. Anyone who partners with us can access [omitted] to open a ticket.

3 Codes:

- _Mechanisms / ● Issue trackers / ● Requirements change identification

17:4 ¶ 66 in P15 INTERVIEW TRANSCRIPT

We also have the cooperation service questionnaire, which is the call [omitted], [omitted], which provides targeted service that they do directly with the technical consultants.

3 Codes:

- _Mechanisms / ● Questionnaires / ● Requirements change identification

17:5 ¶ 66 in P15 INTERVIEW TRANSCRIPT

he opens a ticket at [omitted], and the consultant gets in touch with him and provides support.

3 Codes:

- _Mechanisms / ● Issue trackers / ● Requirements change identification

17:7 ¶ 66 in P15 INTERVIEW TRANSCRIPT

[Omitted] is a system for internal employees and cooperative members to send suggestions for improvement.

5 Codes:

- _Mechanisms / ● Requirements change identification / ● Requirements negotiation / ● Requirements prioritization / ● User feedback tools

17:8 ¶ 70 in P15 INTERVIEW TRANSCRIPT

On our website, there is contact us. It's already happened, and we receive a lot

3 Codes:

- _Mechanisms / ● Built-in user feedback modules / ● Requirements change identification

17:9 ¶ 70 in P15 INTERVIEW TRANSCRIPT

We have already received suggestions on Instagram. We have a communications team that is responsible for obtaining all these demands and directing them.

3 Codes:

- _Mechanisms / ● Requirements change identification / ● Social media platforms

17:10 ¶ 86 in P15 INTERVIEW TRANSCRIPT

I think it's beneficial, but you need to know how to separate and filter this feedback because we receive suggestions for improvement, bug reports, and complaints daily without asking. If you can identify and prioritize this feedback, it can bring good results

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

17:11 ¶ 42 in P15 INTERVIEW TRANSCRIPT

For example, all have access to [omitted], where they can search for ideas or register suggestions. If it's a suggestion we think is viable, we can accept it and leave it open for voting, and then people will vote. Do you understand? So, the suggestions that receive the most votes are chosen

4 Codes:

● _Mechanisms / ● Requirements negotiation / ● Requirements prioritization / ● Voting

18 P16 INTERVIEW TRANSCRIPT

Text document, 2 quotations

P16 INTERVIEW TRANSCRIPT Researcher: I started recording, and we will begin by presenting some concepts that will be important for our interview. The first concept I present to you is that of a software ecosystem. I brought up a concept that I consider very simple: an ecosystem is a group of projects developed and coevolved in the same environment. What would it mean to coevolve in the environment? In general, when we talk about a software ecosystem, we have a common technological platform and software projects developed around this platform. From this, we have a set of actors who interact in this ecosystem; they are part of this ecosystem, which is a central organization, the owner of this platform, and maybe the owner of other projects. Some internal developers are part of this organization. Some external developers are those who develop products and can add products to this platform to benefit themselves because when they make them available on the platform, they can reach other people and work cooperatively with the central organization and other developers to prosper. And there are end users who will use the platform or the projects around it, ok? Interviewee: Okay. Researcher: An efficient example of this is the Amazon ecosystem. It has multiple software products that evolve in the same environment around its platform and have integrations, where external developers can develop an API, some complement to a product, and thrive in that ecosystem. Did you understand? Interviewee: Yes, yes. Researcher: The second concept is requirements management. You probably know this concept. However, I will make it clear in this presentation that I consider requirements management in this research to be the process that follows the entire life cycle of a requirement, focusing on controlling the changes it may undergo. In other words, requirements management has activities related to change management, traceability, version control of requirements, and communication, negotiation, and prioritization of requirements, according to IEEE. Interviewee: Okay. Researcher: The third and final concept is crowd-based requirements engineering. Today, we are experiencing a significant increase in the use of mobile device applications. So, a popular app like WhatsApp or Facebook has millions or thousands of users, and sometimes, you have to think of a new feature. You shouldn't just think about a few of these users but rather the feedback from many users. User feedback is an important resource in modern software development, often containing requirements that help address user concerns and desires for one or more software products. A crowd in software ecosystems is a large heterogeneous user base giving requirements on multiple channels with multiple stakeholders. Crowd-based requirements engineering includes all

approaches that engage a crowd of mostly unknown people to perform required engineering tasks or provide requirements-relevant information. When we listen to a crowd, we consider the concept of collective intelligence. In other words, it hardly makes sense if we analyze a crowd and that crowd is saying something. Did you understand? Interviewee: Got it. Researcher: And what is the role of crowd-based requirements engineering? You can't listen to a crowd manually. For example, I can't interview the users of your scanning application because there are thousands of teachers. So the techniques we used to manage requirements before, like interviews, become more complex, right? Interviewee: Right. Researcher: We have finished the first part. Do you have any questions about the concepts? Interviewee: No, I don't, it's clear. Researcher: So, let's start the second part of the interview. The first question is, have you ever considered a crowd to identify requirements or requirement changes? Interviewee: Got it. So, not directly. I don't know if there is a technical name for this, but how did it happen? Every month, I met with a relationship coordinator between the application and a group of teachers, specifically teachers from [omitted]. And then, when I met with her and a few other teachers who were the school coordinators, there weren't that many people in this case. And they told me how the school teachers were using the app. And they told me that many people cannot scan because of this. I think the answer sheet should have this; I think it could be improved if I could click here; they were talking about this kind of thing, really about the application's usability, and so I was writing it down. At the time, I already had [omitted] the requirements. She wrote down some things, and her team and the usability and UX people formulated them and then passed them on to me. Do you understand if there was any change if they thought it was better this way or changed it to another way? We collected this feedback through questionnaires with teachers who reported to the coordinators. These condemners participated in meetings with us, and we received this feedback from them. Researcher: Let's move on to the next question. Based on what you identified and sent to the UX people, how does the requirements prioritization occur? Interviewee: We defined these requirements, then went back to the teachers to check if everything was ok with these requirements. The prioritization issue was more with [omitted], as he was the representative of [omitted], the project coordinator. He would come directly to me and say, we need this here yesterday, understand? If a requirement had to be implemented; otherwise, it would make the application unfeasible, we would prioritize it. Researcher: Was there an impact analysis? Interviewee: The impact analysis was very subjective. They said what needed to be done, and we planned and delivered the sprint. It was more subjective, and they left it up to me to select what was most critical. Researcher: How did requirements negotiation work in the ecosystem? Did you consider the crowd? Did you consider multiple actors? Interviewee: At the beginning of the projects, I participated a lot in negotiating the requirements, which had to be functional, but as the team grew and the tasks were divided, I was a little left out of this part. Researcher: And what was it like at the beginning? How did this negotiation work? Interviewee: In the beginning, the interview was very informal. For example, how it worked, we had a month in the semester when we went to [omitted], met with the [omitted] team, the [omitted] team, in this case, there was no one, in the majority. Sometimes, there was no one specialized in technology. In this case, it was more stakeholders. And they were the ones who said what needed to be done. And it was

all informal. They did not fill out any formal documentation or anything like that. The principal negotiating currency at the meeting was the question of time. We see that all the time, it appears that things [omitted] were due yesterday. These were features that in software development are not simple to do. We negotiated a lot about the issue of time. Look, it's going to take months, it's going to take weeks, that kind of thing. But there were no conflicting opinions among [omitted] staff. What was conflicting for us was the issue of time, right? We wanted one for yesterday, but we couldn't do that much. Researcher: How did these requirements be communicated? What communication channels do you use to reach the crowd? Interviewee: As I was more present regarding meetings, I saw directly what the client wanted. It was more of an informal conversation. But then, we started having weekly meetings every two times a month. I would meet, usually with [omitted], and she would tell me the client's needs. The documentation part was in Confluence. Researcher: Got it. Have you ever used or know of automated or semi-automated approaches that help manage requirements considering user feedback in a software ecosystem? Interviewee: Automated, no. Researcher: And semi-automated? Interviewee: I don't know if Confluence fits as semi-automated. For requirements management, it is Confluence. Researcher: Do you think that considering feedback from a crowd benefits requirements management in a context where you have multiple actors from different organizations, such as software ecosystems? Interviewee: I think it benefits when many users are involved and it is possible to perform automatic analysis. However, I don't know if this would fit into the scope of [omitted]'s projects because we still don't have much user data to generate a good result. Do you understand? For this type of analysis, it would be necessary for many users to use the applications, so yes like, there would be a good crowd. Researcher: So, in other words, you say that this crowd feedback benefits from having a good amount of data to analyze. Interviewee: Yes. Researcher: So, we ended the interview here, and I wanted to thank you for the time you spent answering the questions here.

2 Quotations:

18:1 ¶ 30 in P16 INTERVIEW TRANSCRIPT

We collected this feedback through questionnaires with teachers who reported to the coordinators. These condemners participated in meetings with us, and we received this feedback from them.

3 Codes:

- _Mechanisms / ● Questionnaires / ● Requirements change identification

18:2 ¶ 62 in P16 INTERVIEW TRANSCRIPT

I think it benefits when many users are involved and it is possible to perform automatic analysis.

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

19 P17 INTERVIEW TRANSCRIPT

Text document, 5 quotations

P17 INTERVIEW TRANSCRIPT Researcher: Okay, the recording started. In part one, the idea is to define a software ecosystem for you. Some people involved in software ecosystems do not know the concept. I brought a definition here that I consider very simple of a software ecosystem, defined as groups of software projects that are developed and co-evolved in the same environment. This environment has a common technological platform accessed by different partners from other organizations and users. Around this platform, there are software projects that co-evolve. In this same environment, I also have a set of actors who relate to both the platform and these projects, which would be the central organization, which is the owner of the platform, the internal developers who are part of this organization, the external developers who are from other organizations or who are autonomous developers who can develop projects for the ecosystem around that platform. The users benefit from the entire ecosystem, both from the platform and these projects. An example I usually give is of Amazon's software ecosystem. As you can see in the image, multiple products co-evolve. It was clear? Interviewee: Yes, you can continue. Researcher: The second concept is requirements management. Requirements management is a process that follows the requirement lifecycle, capturing and mapping the changes that occur in these requirements. Based on this, some activities that interest me and will be in the interview questions related to requirements management are identifying changes to requirements, communication, negotiation, and prioritization of requirements. Ok? Interviewee: Yes, I understand. Researcher: The last concept is crowd-based requirements engineering. This concept has emerged a lot due to our new reality in software development. I always give the example of the Android and iOS ecosystems, where you have several applications used by hundreds of billions of people and where users have different ways of providing feedback, including star rating comments, and this context. There is a large and diverse set of data, including compliments and complaints from many user feedback. User feedback is an important resource in modern software development, often containing requirements that help address user concerns and desires for one or more software products. A crowd in software ecosystems is a large heterogeneous user base giving requirements on multiple channels with multiple stakeholders. CrowdRE includes all approaches that engage a crowd of mostly unknown people to perform requirements engineering tasks or provide requirements-relevant information. That was the last concept. Were all concepts clear? Can we proceed to the questions part? Interviewee: Yes, it was clear Researcher: The first question is, have you ever considered a crowd to identify requirements or requirement changes in software ecosystems? Interviewee: Regarding proprietary ecosystems, Keystone sought the requirements of its partners. She met with partners at partner-only events. They were large companies, there were many people, they were huge events, it was a big money thing. Researcher: What approach did they use to identify requirements from this crowd you mentioned just now? Interviewee: The platform itself had a marketplace that allowed direct contact with users through messages. On the marketplace, there were general comments about the applications or the platform

itself, a star rating. The platform had email contact and the partners could send suggestions via mailing list. Partners could send suggestions, not the users, but the partners, through the forums. Researcher: The partners possibly listened to their users to make these suggestions, right? Interviewee: Certainly, they consulted the market and saw what the platform needed to have. It's interesting to have this to make our work easier. Make these requirements here, right? So this was like a communication channel. And how two of them had a partnership manager. They didn't deal with the crowd itself. They sent the partnership manager to deal with the crowd. However, some partnership models gave the privilege of a more substantial group for the company to have direct access to talk to Keystone about what is essential to change or do differently. One interesting thing I now remember is that they published a roadmap of what they wanted from the platform, which they call a roadmap. They made it available to partners so they could understand more or less where the platform was going, but it went beyond that. It wasn't just for them to understand, for them to converge the solutions, it was also for them to give suggestions, look, it would be interesting to have more that. So, they make the roadmap available for that classic issue that they will direct in this direction and for yours so that we can have compatibility. No, it wasn't just in that sense; it was to obtain suggestions, too; in addition to this direction, the Keystone does this more here. And they would then work on this roadmap, close it, and say, ready, this will be the company's management, obviously not giving access to the organization's strategic, let's say, sensitive information. Making the roadmap available also helped to obtain requirements or changes to requirements. It was not just about telling the direction in which the platform was going, but it was also about getting suggestions for direction for the platform, and this was about partners. Researcher: The next question is more related to the prioritization of these requirements, so they listen to partners, whether through face-to-face meetings, through the roadmap, or the tool. Did you notice if they prioritized considering these partners, this crowd, or, for example, if they prioritized considering this smaller group that the partnership manager organized? If this prioritization occurs, what approaches did they use? Interviewee: Look, it's an interesting approach, it's the case that most talked about this issue of obtaining requirements, Keystone is just one company. He does the following: he goes to the market, he calls it the customer's pain, doing market research, and that whole administration issue. Market research, see the market need, and then given that he saw this market need, he said, right, now who could put this on my platform, which I don't have yet? Then, among those partners who made suggestions, he checked who could solve his problem. Then, he already had a prioritization and knew what the market wanted. He had already prioritized what they wanted to be solved, so when they got the partners, it was already for resolving previously prioritized things. Not from the partner's requirements, he wanted the solution, so he took it from the market, the market's decisions, or sometimes the needs of their customers, who said, look, we need this, that would be interesting. So he had the market need they saw through market research, for example, market analysis and the need of customers who had already exposed the need to them. The prioritization was not based on the partner; it was based on what they needed for their client or for a need they already had, or for the market they wanted to enter or expand, new things market. Partners only came in to solve, so they asked partners to solve some solutions based on the

prioritization they already had of the requirement. They went to fairs, for example, for startups. They then checked to see if there was someone there who could solve it when they hadn't found it with their partners. Researcher: And since he had prioritization, let's move on to the third question, which is about negotiating requirements. There is a proximity between prioritization and negotiation. I'm differentiating this by saying that negotiation occurs when we have conflicting requirements. That is, I may have requirements from different partners that cannot be accepted. Because if I take one, I invalidate the other. This requires negotiating requirements and discussing with partners to reach an agreement. Interviewee: It would be exclusive, wouldn't it? I don't remember them mentioning conflict management. They even said that we had to manage conflicts between different requirements, that sort of thing. However, they did not provide details, not least because of the direction of the research. Details of how to negotiate conflicting requirements were not addressed. Researcher: Got it. The next question is related to requirements communication. How did requirements communication occur in this ecosystem, considering the multitude of users and partners? What channels were used? Interviewee: Communication, as I told you, was more closed. They had approaches that were not directly related to communicating requirements. They would analyze the market, do market research to see the market's needs, it was not someone I was going to tell them the requirements, it was that more classic thing, I go to the market to see what they want, in short, market research. And besides that, no one talked about analyzing the data generated on the platform and forums. Do you understand? None of them mentioned, for example, they even make it available, I believe they do, because in their marketplace, there is the option to evaluate the applications, to make suggestions. So, I believe they did it through the suggestion options and the comments, right? On the marketplace, general comments about the application or the platform itself, the little evaluation with a few stars. Researcher: Do you know of automated or semi-automated approaches to managing user feedback or the requirements coming from this crowd in the context of software ecosystems? Interviewee: You can mine data, mine information, requirements, so to speak, from your customers. And what do they want? Which they possibly wish to do. They develop applications related to innovation. They created an internal tool to retrieve all comments and analyze the trends of the application itself and the ecosystem. Imagine his keystone if it can explore a little of what you said about the roadmap, right? If it can look at trends in what we are talking about in the different applications there, it can also analyze the trend of the ecosystem itself. If he's evolving, where he's going, he's going more niche. If he is going more to someone else and to whoever he is a partner, he can also know the trends of his tool, his product. Researcher: The last question is whether you consider that taking into account crowd feedback benefits requirements management in a software ecosystem. Interviewee: It is interesting, I think it benefits. Let's assume we can analyze the texts from the forum or some area disponible in the ecosystem platform to make suggestions. We will have suggestions from each client and partner. You can get the requirements patterns of these partners. You can catch the feedback and trends of what customers want. It's that question of him taking the crowd's wisdom and using it to identify trends. Focusing on the customer, in short, you can get information that helps you focus on your customers. So, we can collect information about

requirements, relevant requirements, and even innovative ideas. As one of those I interviewed in this area said, he said the following: always look at the customer, always look at the customer. Hence, the focus is on the customer, generating value for the customer. So, considering customer focus and generating customer value, they unanimously said it allows you to grow the company. You are thinking that the client is giving his suggestions there, and from what I know about data mining, right? What can you generate in terms of results, so can you get ideas you often haven't even thought of? Researcher: Considering software ecosystems, these customers can be divided into different levels, right? Interviewee: It's interesting because think about the following: if you're on the platform, and let's assume we're going to analyze the texts on the forum or in some area that you post to make suggestions, you have suggestions from each client of each partner of yours that has a solution. And the solution in general for your platform, so you see that by doing data mining, you can get information about requirements, relevant requirements, and even, as I said, innovative ideas that you haven't thought of or suddenly a partner hasn't thought of. Boy, this is interesting; depending on the strategic level of importance of the new idea, you implement it and put it on your platform. Do you retain the idea of greater value, or would you pass it on to a partner, or would you look, as I said, for someone else who could be a partner who would have a suitable solution there? Researcher: [omitted], we have reached the end of the interview. Thank you for your time. Thank you again for your contribution to my research. I think the interview was fascinating.

5 Quotations:

19:2 ¶ 22 in P17 INTERVIEW TRANSCRIPT

The platform itself had a marketplace that allowed direct contact with users through messages. On the marketplace, there were general comments about the applications or the platform itself, a star rating

3 Codes:

● _Mechanisms / ● App store / ● Requirements change identification

19:3 ¶ 22 in P17 INTERVIEW TRANSCRIPT

Partners could send suggestions, not the users, but the partners, through the forums.

3 Codes:

● _Mechanisms / ● Forums / ● Requirements change identification

19:4 ¶ 39 in P17 INTERVIEW TRANSCRIPT

On the marketplace, general comments about the application or the platform itself, the little evaluation with a few stars.

3 Codes:

● _Mechanisms / ● App store / ● Requirements change identification

19:5 ¶ 47 in P17 INTERVIEW TRANSCRIPT

It is interesting, I think it benefits. Let's assume we can analyze the texts from the forum or some area disponible in the ecosystem platform to make suggestions. We will have suggestions from each client and partner. You can get the requirements patterns of these partners. You can catch the feedback and trends of what customers want. It's that question of him taking the crowd's wisdom and using it to identify trends.

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

19:6 ¶ 22 in P17 INTERVIEW TRANSCRIPT

The platform had email contact and the partners could send suggestions via mailing list

3 Codes:

- _Mechanisms / ● Mailing lists / ● Requirements change identification

20 P18 INTERVIEW TRANSCRIPT

Text document, 5 quotations

P18 INTERVIEW TRANSCRIPT Researcher: I started recording here. So, I'm going to start introducing some concepts to you. The first concept is that of software ecosystems. I brought up an idea that I consider the simplest so that the people I am interviewing in the market can understand this concept. Often, these people are contributing to a software ecosystem, but they don't know the definition of it. A software ecosystem is a group of projects that are developed and co-evolved in the same environment. What would this same environment be? For you to understand, we have a common technological platform, and around this platform, we have projects that evolve. Furthermore, in an ecosystem, there is a diverse set of actors, such as the central organization, which is the owner or responsible for the platform, and the internal developers of that organization, as I made clear in the image. Some external developers develop these projects around the central platform. And users, too, ok? Interviewee: Okay. Researcher: To illustrate this concept, I brought an example—the Amazon software ecosystem. Amazon had a platform yesterday, and several other products, like Prime and AWS, were evolving in the same environment. Right? Interviewee: Yes Researcher: The second concept is requirements management. Requirements management tracks requirements throughout their lifecycle. In other words, it is concerned with the definition of these requirements but mainly with the changes that may occur in these requirements over time. Whether changes in the business area or technical changes. Requirements management activities related to change management, version control, traceability, communication, negotiation, and requirements prioritization. Did you understand? Interviewee: yes Researcher: The third and final concept is crowd-based requirements engineering. It is a concept that has also been discussed in requirements

engineering. It includes all approaches that engage a crowd of primarily unknown people to perform requirements engineering tasks or provide requirements-relevant information. A crowd in software ecosystems is a sizeable heterogeneous user base giving requirements on multiple channels with multiple stakeholders. User feedback is an essential resource in modern software development, often containing requirements that help address user concerns and desires for one or more software products. The example I have presented is that of mobile device app stores that allow users to send feedback through stars and comments. Depending on the application's popularity, this feedback can be sent by hundreds, thousands, or millions of people. Based on this feedback, an analysis can be carried out to identify changes, new requirements, and trends for that product. However, doing this analysis manually is complicated. Due to the amount of data, it is costly and does not often bring you a return. So crowd-based requirements engineering also aims to help in this context by reading and analyzing this feedback automatically or semi-automatically. These were the concepts, and I wanted to know if you understood them. Interviewee: Yes, I understand! Researcher: I will ask you some questions about these concepts starting part two of the interview. The first question is, have you ever considered crowd feedback to identify requirements or requirement changes in your software ecosystem? Interviewee: So, come on, we usually receive this feedback from an intermediary in the crowd. A person who received this information conveyed to us how the crowd wanted the requirements. Yes, we have that. Researcher: Do you know how these requirements and ideas usually got to her? Interviewee: Generally, customers go to physical stores, and there are ways for them to express their opinions. We also have some processes here to get to know the customer, where the person also gets to see the customer, but then it's not so much about the customer's opinion but the customer's profile. We are starting to use WhatsApp channels now, and people are really sending their opinions through them. Before, it was much more closed because it was a system in which you had to go to [omitted] to be able to carry out any negotiations, whether to grant credit or return contracts, on a day-to-day basis. Still, today, we already have this audience from WhatsApp and the [omitted] website and they can give an opinion. Researcher: Do these opinions come directly to you, or do they reach this intermediary? Interviewee: No, they arrive for each management area. We also have customer service where some situations arise, but when they pass the requirements on to us, they sometimes pass with the customer's perception. It's rare, but it happens; the customer has this perception here. We have to find a solution. Taking a customer perspective here as a basis, it has already happened. Researcher: Now, considering these demands and requirements, The next question is, how do you prioritize what will be done? Is there, for example, a response from the person who asked to know, suddenly more things to see what is more priority or the conversation between you and this manager who passes by, how does prioritization occur with the crowd? Interviewee: So, prioritization is up to the manager, we don't have that freedom, I'll put it that way, to say what is a priority or make a prioritization list. Several factors carry out this prioritization. Sometimes audits, sometimes through legal measures. Researcher: Prioritization can have a financial impact, right? Interviewee: It could have a more significant financial impact, as I work in one [omitted], and it could also be political. Politics puts it as paramount. For example, a while ago, people ran a campaign and called it [omitted].

So, we had to work on our system to achieve the objective set there by the strategic area, coming to the political area. So, I'm not going to say what it was, right? But it was to pay interest, lower interest on a type of loan, so that's it. The modality didn't exist; we even had to create the loan modality to adapt the system to what was requested. So this came, for example, ahead of the other demands. We sometimes have external or internal audits when demands take priority over others. There are legal demands. Sometimes, court orders arrive that we have to act in light of what has been determined by the legal system and the court. And some demands come directly from [omitted], right? They also go ahead of the others, so, on a day-to-day basis, apart from these situations, on a day-to-day basis, prioritization is done according to what the manager determines as a priority. At least here, we don't have much to say in this situation. Researcher: The next question focuses on negotiating these requirements. You say that they already arrived prioritized for you, right? You have also shown several examples of how this prioritization can occur. When we talk about negotiation, we focus more on issues of conflicting requirements. Maybe in the banking context, this works; I can even give an example: often, a manager works with several managers, but manager A asks for something, and manager B asks for something that is the opposite of that; that is, you can't answer both. You will have to negotiate with them. Does this situation occur, and how does this negotiation with the crowd occur? Interviewee: It doesn't happen at the level of one being contrary to the other. It happens that everyone wants to be a priority. Then it happens. If one is against another like that, this doesn't happen. It is a very well defined product, it has the credit granting area, the bidding area, and the object storage area. So everyone has their area, so what happens? One doesn't go too far to contradict the ideas in the other's area. Each one is responsible for their area, but everything converges on the product owner. They all come together with the product owner when deciding which demands will enter the sprint. To be able to be contrary to someone else's idea, I don't remember it happening. Researcher: When you consider this crowd, right? You say it doesn't come directly to you. It comes through these managers, right? But how does communication occur? Interviewee: This happens through customer service system. Researcher: Considering the crowd, the next question is whether you know or have used an automated or semi-automated approach to manage these requirements. Interviewee: Not automated using artificial intelligence. We have requirements control tools here. Here, we use an IBM framework, GIT, which includes the project part, the demands, and the prioritisations. The requirements part is what we call RDMG. And there is the quality part, which is the RQM. It has versioning and the ability to monitor all functional and non-functional requirements. Researcher: So, thinking about tools, using more interesting, current approaches. Do you consider listening to this crowd benefits requirements management in the software ecosystem? Interviewee: I see that it helps more than it hinders. Because they are the end consumers, we work for them to use it on the tip. So, listening to this crowd and bringing it into the context of the requirements helps. Researcher: I'll adjust the question a little. Do you think it would be more effective to listen to a small group of users, more representative, as we usually do at the end, suddenly, or try to listen to all these users and have some way of analyzing this in a more automated way? Interviewee: Listening to this crowd and bringing feedback into the requirements

context benefits requirements management. Researcher: Well, [omitted] we arrived at the end of the interview. Thank you very much for your time.

5 Quotations:

20:1 ¶ 26 in P18 INTERVIEW TRANSCRIPT

We are starting to use WhatsApp channels now, and people are really sending their opinions through them

3 Codes:

- _Mechanisms / ● Requirements change identification / ● Social media platforms

20:2 ¶ 26 in P18 INTERVIEW TRANSCRIPT

Before, it was much more closed because it was a system in which you had to go to [omitted] to be able to carry out any negotiations, whether to grant credit or return contracts, on a day-to-day basis.

3 Codes:

- _Mechanisms / ● Built-in user feedback modules / ● Requirements change identification

20:3 ¶ 26 in P18 INTERVIEW TRANSCRIPT

Still, today, we already have this audience from WhatsApp and the [omitted] website and they can give an opinion.

3 Codes:

- _Mechanisms / ● Requirements change identification / ● Social media platforms

20:4 ¶ 46 in P18 INTERVIEW TRANSCRIPT

This happens through customer service system.

3 Codes:

- _Mechanisms / ● Requirements change identification / ● User feedback tools

20:5 ¶ 58 in P18 INTERVIEW TRANSCRIPT

Interviewee: Listening to this crowd and bringing feedback into the requirements context benefits requirements management.

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

21 P19 INTERVIEW TRANSCRIPT

Text document, 4 quotations

P19 INTERVIEW TRANSCRIPT Researcher: I started recording. As I was telling you, the first part of the interview aims to introduce you to some concepts that are important for the questions I will ask you soon and also helps us check if we have the same definition for them. OK? Interviewee: Okay. Researcher: You already know the first concept. So, I'm going to go through it a little faster. I'm using a straightforward definition of software ecosystems, that of Lungu. Lungu defines SECO as groups of developed and coevolved projects in the same environment. I have presented an example that shows the common technological platform, the projects around this platform, and some actors that I differentiate between the central organization, external developers, and users. To make it easier to understand, I give the example of the Amazon ecosystem and the example of the [omitted] ecosystem because I am interviewing some people there. Amazon has multiple software products that evolve in the same environment around its platform and have integrations, where you have external developers who can develop an API, some complement to a product, and thrive in that ecosystem. Did you understand? Interviewee: Understood. Researcher: The second concept is requirements management. Requirements management aims to monitor and control the requirement throughout its life cycle, focusing on the changes that may occur in that requirement. So, from identifying a change, how it affects the requirement, and how that requirement transforms. Thus, requirements management relates to change management, traceability, version control, communication, prioritization, and requirements negotiation. OK? Interviewee: Okay. I understood. Researcher: The last concept is crowd-based requirements engineering. The idea is to use semi-automated or automated approaches to collect information from a large and heterogeneous number of users, a crowd. A crowd in software ecosystems is a sizeable heterogeneous user base giving requirements on multiple channels with multiple stakeholders. Furthermore, user feedback is an essential resource in modern software development. The example I have presented is that of mobile device app stores that allow users to send feedback through stars and comments. Depending on the application's popularity, this feedback can be sent by hundreds, thousands, or millions of people. Based on this feedback, an analysis can be carried out to identify changes, new requirements, and trends for that product. However, doing this analysis manually is complicated. Due to the amount of data, it is costly and does not often bring you a return. So, crowd-based requirements engineering also aims to help in this context by reading and analyzing this feedback automatically or semi-automatically. These are the three concepts that were on the form. Were they clear? Do you think you fit into this context? Interviewee: I understand them better now. I believe I am included. This multitude of users has more to do with how we analyze, too, right? Researcher: Yes. It could be, too. So, now we can start with the second part of the interview, where I will ask you some questions related to these concepts. The first question is if you consider or have you considered a crowd, which we can think of with a large heterogeneous set of users or even other stakeholders who can take on the role of users to identify changes in requirements in software ecosystems. Interviewee: I understand. I observed this in a research on [omitted] and [omitted]. Because [omitted] has a powerful competitor, [omitted], right? So, users make many suggestions to meet all the features that [omitted] does. Even they also enter these requests. Because they have a comparison between the wikis and documentation, we do this about [omitted], and if you are a

user of [omitted], you can migrate to our software. It's like an adaptation manual, so we are concerned about competing with them. So, users ask for many things available in this other software commercial, which is expensive for this free software, which is from the [omitted] community, the [omitted]. They ask for a lot of things, so there's a reason why [omitted] they do it like that, so it would be good if it was here and done like that. So, I realized that they respond, the people who develop it say it will be done, or that they have already done it, or that it is in the development process. So, how does the community itself develop the features? Like, it's in the queue for development. They would submit this feedback on the project's community forum [omitted]. They respond to each other in several specific forums of the community. Some people were critical in the community. They were the main ones. For the answers they were giving. It's always those people who responded. I realized they were the most active and active developers, but I don't know how they analyzed it and what criteria they used very well. Researcher: What about considering crowd feedback when prioritizing requirements? Did you think it? Interviewee: As it was an open-source community, we did not get significantly involved in this prioritization process. A more active group of people ended up directing the community. Researcher: And did they negotiate with you to implement any changes? Interviewee: No, we could request it on the forum and wait for a response. Either it was a justification for not doing it, or it was saying that it had already been done, or that it was in the process of being implemented. Researcher: This communication with the crowd took place on the forums. Were there other channels where crowd feedback could be collected? Interviewee: There was communication in annual meetings and in these bug trackings. They had conversations on [omitted] right? Also, meetings to discuss this and discuss the work. And so some open-source communities put together a list of things to do, you know? There was a community that put together a list of things to be done. So, if I wanted to start participating in [omitted], for example, I want to start contributing, where do I start? And there were some more manageable things. And you chose, right? But I think it was on GitHub; now I don't remember which one, and there were some minor things. Others had mentoring programs. They have some programs where they have mentors, and you could participate and sign up; if you were approved, you had a mentor for a project, and this mentor guided you and gave you some things to do. Now, what he was going to do, how, what the requirement is, what the contribution is, I don't know how they choose. There is a program called [omitted] that free software communities write themselves. The [omitted] is one of them. The students also wrote to themselves, the people who wanted to participate. If they were approved, they provided, if I'm not mistaken, features to be developed together with the mentor. The person developed those features with a mentor's guidance, which was a way for them to start engaging in the community. Researcher: The next question is if you know or have used any automated or semi-automated tool to identify or analyze this crowd feedback in the context of the previously mentioned communities, which are software ecosystems. Interviewee: I didn't use it and don't know it. I didn't see any mention of the tool within any of these communities. Researcher: The last question is whether you consider that feedback from a crowd benefits requirements management in a software ecosystem, considering the concepts I presented at the beginning of the interview. Interviewee: I think it is beneficial. I think it's important to listen to the crowd

because they are the ones who use the systems and know their needs. I always use an example when I teach software engineering that will fit this question well: the following. I went to [omitted] and changed the password, but the manager could not update it. And then she asked about my profession. I said she was a systems analyst, at the time she wasn't a teacher. She told you systems analysts change everything. Before, I knew how to do everything in the system; now, you change everything, and we no longer know how to do anything. It has to be changed, it changes the whole system, it disrupts our entire service. I got scared. I started to think like this: whoever develops the systems at [omitted] is in [omitted] and is not going to ask the manager here at [omitted] how she works. And that's exactly it, this question of understanding how the manager here at [omitted] works. They won't ask the whole [omitted]. So, what will happen? They will find what is best there for them, they change the system there and here in [omitted], she changes her work, disrupts everything, really for her it is not the best way to do it and for many people around of [omitted], it is not either. So, that makes it difficult. There is this issue of getting a sample and someone from some cities. But you won't serve everyone, right? But it's a matter of whether you listen. The more people you can hear, the better you will be able to serve them. And she spent almost half an hour trying to change the password. She said that she had to enter one screen, enter another, change here, change there and that it wasn't like that before; the whole system had changed. And she was angry with the system analysts because of this change. Researcher: Got it. Your example is perfect and has much to do with my research. [omitted], we have reached the end of the interview. Thank you for accepting the invitation and asking if you have anyone to recommend.

4 Quotations:

21:1 ¶ 21 in P19 INTERVIEW TRANSCRIPT

They would submit this feedback on the project's community forum [omitted]. They respond to each other in several specific forums of the community

3 Codes:

- _Mechanisms / ● Forums / ● Requirements change identification

21:2 ¶ 33 in P19 INTERVIEW TRANSCRIPT

There was communication in annual meetings and in these bug trackings

3 Codes:

- _Mechanisms / ● Issue trackers / ● Requirements change identification

21:3 ¶ 33 in P19 INTERVIEW TRANSCRIPT

They had conversations on [omitted]

3 Codes:

- _Mechanisms / ● Requirements change identification / ● Social media platforms

21:4 ¶ 41 in P19 INTERVIEW TRANSCRIPT

I think it is beneficial. I think it's important to listen to the crowd because they are the ones who use the systems and know their needs

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO

22 P20 INTERVIEW TRANSCRIPT

Text document, 4 quotations

P20 INTERVIEW TRANSCRIPT Researcher: I started recording, and now we'll begin the interview part, in which I'll introduce you to some important concepts. In this part, I will begin by presenting the concept of software ecosystems, which can be defined differently. I brought a definition that I consider the simplest. So, we can understand a software ecosystem from a project perspective, where groups of software projects are developed and co-evolved in the same environment. In other words, they have some connection, and one project can help the other evolve. To make this more precise, I brought an example of a common technological platform where projects are developing around this platform, using this platform to communicate, integrate, or offer more services. Some actors from this set of platforms and projects are involved in this software ecosystem. A central organization that has control of the expected technological platform, internal developers who are part of the central organization, external developers who can work on projects around the platform, and users who benefit from the results of these projects. Right? Interviewee: Okay. Researcher: A practical example of this is the Amazon ecosystem, where we have a central Amazon platform, and some products evolve around this platform that have some integration with it, such as Amazon Prime, Amazon Music, and AWS. Do you know? Did you understand the concept? Interviewee: Yes, yes. Researcher: The second concept is requirements management. We usually define requirements management as more focused on the changes a requirement may have throughout its life cycle. It is the one that accompanies, captures, and maps from the origin to when that change will be executed. So, when talking about requirements management in general, we are concerned with managing requirements changes. Other activities related to requirements management are requirements communication, negotiation, prioritization, status monitoring, version control, and traceability. Is that okay? Interviewee: Ok! Researcher: The last concept is a concept related to user feedback. User feedback is an important resource in modern software development, often containing requirements that help address user concerns and desires for one or more software products. The concept is that of crowd-based requirements engineering. This area of requirements engineering includes all approaches that engage a crowd of mostly unknown people to perform requirements engineering tasks or provide requirements-relevant information. A crowd in software ecosystems is a large heterogeneous user base giving requirements on multiple channels with multiple stakeholders. Crowd-based requirements engineering considers a crowd of users and potential users to automatically or semi-automatically identify and analyze their feedback. These were the concepts I had

already presented in the questionnaire. Do you already know them? Do you think you fit into this context? Interviewee: Yes, I had seen it in the questionnaire, but now explaining it here is more apparent. Researcher: Now, moving on to the questions. I will ask you some questions related to your experience in software ecosystems, taking into account the concepts I presented to you. The first question is, have you ever considered a crowd, this heterogeneous set of users and potential users, to identify requirements or changes to requirements? How do requirement changes arise? Interviewee: What would be a large set? How many people are in the organization? In my organization, it is a very select group that proposes changes to product requirements. They are usually business managers for that product. We are talking about executives. Some executives have more ideas, but we talk about a tiny group. Researcher: And do the products you offer have many users? Don't these users demand changes? Don't they complain about bugs? In general, bugs usually lead to changes in requirements. Interviewee: Perfect. Not always new features or changes to business rules. Researcher: Exactly. We believe that some requirements will change, but it may not. Identifying changes to requirements often comes from this group of users, which in your case must be thousands of people. Do you understand? Interviewee: Yes. Researcher: Do you consider these thousands of users when identifying requirements? Interviewee: Yes, we do interviews. In our e-commerce, the main user is not the end customer. It is the broker. So, these demands you are talking about for changing requirements and new functionalities generally emerge through these [omitted]. If they want to send feedback, they use the feedback channel of our e-commerce. Researcher: Which is a large group of people, correct? Interviewee: Very big, thousands. Do you understand? And these brokers we do an immersion with the leading brokers. For those who sell the most and those who are most productive, the insurance company takes them for a week in a farm hotel and gives them lots of treats to attend these events. For example, there was already [omitted]. I think it was before the pandemic that they went to [omitted] and spent a week there. Researcher: And are these brokers independent, or are they employees? Interviewees: They are independent. The vast majority sell to another insurance company as well. They bring ideas from outside, too. We call this Congress, combined with interviews. Researcher: Is there any online way for these brokers to report, for example, a bug or request for changes to requirements for the platform or a product? Interviewee: No. It's not formal, such as a questionnaire. If he wants to send something, he uses the chat within the e-commerce itself. Researcher: When a broker uses the chat to request a change, does he speak to an attendant, and does this request come to you? Interviewee: The request reaches the business manager. It does not get the IT sector directly. First, the business manager evaluates. Researcher: Let's move on to the second question. Have you considered this crowd of brokers or another crowd to prioritize these requirements? How does prioritization work? Interviewee: Yes. We consider the market's vision and what the market wants. Who is the market? It's the brokers. What are the brokers asking for most? What are they complaining about most? Researcher: And what kind of approach do you use to prioritize this? Interviewee: They have a tool, but it's also manual, a spreadsheet. Researcher: The third question is whether you have considered this crowd to negotiate the requirements or changes to it. Note that negotiating is different from prioritizing. Generally, when we talk about negotiation, it is because of conflicting

requirements or changes. It's not a case of knowing which demand you will meet first, but instead having the idea that you won't meet another if you meet one. Thus, you need to reach an agreement, either with the central management of the ecosystem or with the crowd itself. How do these negotiations occur in your context? Interviewee: This question is very complex because once the business area understands that that functionality makes sense, that complaint will gain weight. The project demands this, and the negotiation is with IT. Well, what will I prioritize? Imagine that the business manager wants everything at the same time. So the prioritization is with development, it is not prioritization with the requester or the complainant. He probably uses an ROI—a return on investment, which will give more visibility, sometimes a return. Researcher: This relates to prioritization and how negotiating conflicting requirements or changes works. Interviewee: I don't participate and I don't know if there is a technique or method that they use to carry out this type of conflict management negotiation, I don't think there is. It doesn't have that. It's not that kind of collaboration. I've never heard of a partnership like this. I've never heard of it. It's by decree, do you understand? There was the requirement; he understood it was valuable, so let's do it. It was over, and he decreed it. Researcher: The next question is, how do you get this feedback from the crowd? What communication channels do you have with the crowd that allows you to identify changes or even communicate requirements to that crowd? Interviewee: As I said, we have congresses, but changes don't always come from there. They also come through an executive presentation. There is a tool that registers these demands. Researcher: Have you ever used any automated or semi-automated approach to managing requirements in the context of your ecosystem? Interviewee: I've never used it, I saw you explaining it, I thought it was fantastic. It would be really cool if we could make a gear to mine and deliver a result. Researcher: The last question, [omitted], is whether you consider this crowd feedback to benefit requirements management activities in your software ecosystem. Interviewee: Of course, brother. Look at what you just said. Imagine having to do this manually or on demand, right? As someone complains or people complain, you register. Then, at a certain point, you analyze and verify that you see some change several times. This manually isn't good if you have a tool to do it for you. It's perfect, it would be ideal. Increasingly, I think the trend is for this to happen. Hearing user feedback is very good. I consider it beneficial. Researcher: OK, [omitted]. These were the questions I wanted to ask you. Thank you, and if there is anyone you think could help me by participating in the interview, please let me know. I'm going to end the recording. Interviewee: Okay, I'll check.

4 Quotations:

22:1 ¶ 33 in P20 INTERVIEW TRANSCRIPT

So, these demands you are talking about for changing requirements and new functionalities generally emerge through these [omitted]. If they want to send feedback, they use the feedback channel of our e-commerce

3 Codes:

- _Mechanisms / ● Built-in user feedback modules / ● Requirements change identification

22:2 ¶ 45 in P20 INTERVIEW TRANSCRIPT

If he wants to send something, he uses the chat within the e-commerce itself.

3 Codes:

- _Mechanisms / ● Built-in user feedback modules / ● Requirements change identification

22:3 ¶ 69 in P20 INTERVIEW TRANSCRIPT

There is a tool that registers these demands.

3 Codes:

- _Mechanisms / ● Requirements change identification / ● User feedback tools

22:4 ¶ 77 – 78 in P20 INTERVIEW TRANSCRIPT

Of course, brother. Look at what you just said. Imagine having to do this manually or on demand, right? As someone complains or people complain, you register. Then, at a certain point, you analyze and verify that you see some change several times. This manually isn't good if you have a tool to do it for you. It's perfect, it would be ideal. Increasingly, I think the trend is for this to happen. Hearing user feedback is very good. I consider it beneficial.

1 Codes:

- Benefits of user feedback from a crowd in requirements management in SECO