

# Session-1 Transcribed

L: Author 1

E: Author 2

U: Author 3

0:00 - 1:35

L: Öncelikle herkese geldiği için çok teşekkür ederiz. Bugün aynı şekilde bu oturumun kaydedilmesine izin verdiğiniz için teşekkürler. Bugün genel olarak proje sırasında ortaya çıkan yazılım eserlerinin (software artifacts) görsel bir şekilde analizinin ve bu bağlamda alınacak kararlarda destek sağlayacak Software Artifact Analyzer (SAA) adlı aracı size tanıtmak ve en önemlisi sizin hem araç hakkında hem de genel olarak düşüncelerinizi öğrenmek için toplandık.

L: First of all, we would like to thank everyone for coming. We also appreciate your permission to record this session today. Today, we are gathered to introduce you to a tool called Software Artifact Analyzer (SAA), which visually analyzes software artifacts that emerge during the project and supports decision-making in this context. Most importantly, we are here to learn your thoughts and feedback about both the tool and the general topic.

1:35 - 2:16

L: İsterseniz biz bilgisayarlarımızdan görünen wifi'ye bağlanabilirsiniz. Bu sırada ben de kısaca bugün bu çalışmada yapacaklarımızı özetlemek istiyorum. İlk olarak bu haftanın başında sizinle paylaşmış olduğum ön ankette verdiğiniz cevaplar üzerinden gideceğiz. Onlar hakkında konuşacağız. Daha sonrasında ben genel olarak bu projede kullandığımız konseptler hakkında kısa bir anlatım yapacağım. Ardından size SAA aracımızın küçük bir demosunu yapacağım ve üzerinden bir anlatım gerçekleştireceğim.

L: If you would like, you can connect to the visible WiFi from your computers. Meanwhile, I would like to briefly summarize what we will be doing in this session today. First, we will go over the responses you provided in the pre-survey that I shared with you at the beginning of this week. We will discuss them. After that, I will give a brief presentation on the concepts we use in this project. Following that, I will give a short demo of our SAA tool and provide an explanation.

2:16 - 2:51

L: Daha sonrasında dediğimiz gibi sizden bu SAA aracını denemenizi isteyeceğiz ve bu bazı yapmanız için size 2 tane küçük görev vereceğiz. En sonunda da aslında hem bu çalışma boyunca ki deneyiminizi hem de genel düşünce ve görüşlerinizi paylaşmanız açısından sizi bazı sorular yönlendireceğiz ve bu sorular bağlamında da tartışmanızı isteyeceğiz. Bu şekilde şimdi o zaman isterseniz ön anketle devam edelim. Bu arada ana amacımız sizin düşünce ve görüşlerinizi öğrenmek. O yüzden istediğiniz bir noktada istediğiniz gibi katkı verebilirsiniz.

L: Afterwards, as we mentioned, we will ask you to try out the SAA tool, and we will give you two small tasks to complete. At the end, we will ask you some questions to share your overall experience during this study as well as your general thoughts and opinions. We will then ask you to discuss these questions. Now, if you are ready, let's continue with the pre-survey. Meanwhile, our main goal is to learn your thoughts and opinions. Therefore, you can contribute at any point and in any way you like.

2:51 - 3:41

L: Şimdi ilk tur olacağı için ilk başta kendinizi tanıtırıp başlarsanız diğer katılımcıların da sizi tanıması ve herkesin birbirini daha iyi tanıması açısından.

L: Since this is the first round, please start by introducing yourself so that other participants can get to know you and everyone can get to know each other better.

L: Pre-Survey\_1-2.Soru: İlk başta aslında hangi VCS ve Bug Tracking sistemlerini kullandığınızı sormuştuk size. Yanıtlara göre zaten buraya katılan herkes git tabanlı bir sistem kullanıyor.

L: Pre-Survey\_1-2. Question: Initially, we asked you which VCS and Bug Tracking systems you use. According to the responses, everyone here uses a git-based system.

L: Bu noktada işte ben bu aracı kullanıyorum çünkü şu şu yüzden diye bir açıklama yapmak isteyen konuşmak isteyen olur mu acaba?

L: At this point, would anyone like to explain why they use this tool and share their reasons for doing so?

(Cevap Yok) L: Tamam geçiyorum.

(No Response) L: Okay, I'll move on.

3:41- 5:35

L: Pre-Survey\_3.Soru: Daha sonraki sorumuzda ise size daha önce hiç görsel bazlı bir yazılım uygulaması kullanıp kullanmadığınızı sormuştuk. 4 tane katılımcımız kullandıklarını beyan etmiş. Bu kişiler konuşmak ister mi acaba hangi amaçla kullanıyorlar veya hangi araçları kullanıyorlar?

L: Pre-Survey\_3. Question: In our next question, we asked whether you have ever used a visual-based software application before. Four participants stated that they have. Would these individuals like to speak about what purposes they use them for or which tools they use?

Katılımcı-3: Ben bu soruyu çok anlamadım, görselden kasıt nedir burada?

Participant 3: I didn't quite understand this question. What do you mean by "visual" here?

L: Hocam genel olarak bir şeyi yani nasıl dersem aslında hani belli diagramlar içeren veya grafik içeren herhangi bir araç olarak söyleyebilirsiniz. Yani genel olarak proje süresince yararlandığınız bu şekilde bir diyagram, bir görsel element içeren bir aracı kastediyoruz.

L: Generally speaking, you can think of it as any tool that includes certain diagrams or graphics. We are referring to any tool that includes a diagram or visual element that you have used throughout the project.

Katılımcı-3: Evet.

Participant-3: Yes.

Katılımcı-1: Yani ben de evet diyenlerdenim işte bir şeyle beraber

Participant-1: So, I am one of those who said yes, and along with something.

E: Hocam bir de kendinizi tanıtır mısınız?

E: Could you also introduce yourself, please?

Katılımcı-1: Günaydınlar tekrardan ya. Şeyde uzun dönem Havelsan'da işte yazılım mühendisliği, yazılım mimarlığı işte proje yönetimi, scrum masterlık, mimarlık falan genelde işlemler yaptık. Bir ara Havelsan'ın yazılım geliştirme altyapısına bir el atmıştık Eray hocamla beraber o zamanlar çalışmıştık. Uzun dönemde şeylere bu araçları kullanma fırsatımız oldu. Hatta bütünleşik olarak da nasıl yapılabilir diye kaygıya düşmüştük. Fırsatlarımız olmuştü. Buradaki sorunun tabii kapsamı tam açık değil. Yani görselliğin kademeleri çok çok farklı olabilir. Bende işte şimdi mesela en son aldığım Azure DevOps'la beraber kullanırım. Kendiler hem work itemları için hem de source control itemları için böyle normal ilişkisel görsel var ama kastettiğimiz görselliğin gücüyle kabiliyetini tam kestiremediğimiz için evet, bak görselleştirme yapıyorlar. Yani izleyebiliyorsunuz, navigate edebiliyorsunuz birisinden diğerine geçebiliyorsunuz falan işleri takip edebiliyorsun bunu sağlıyor iyi kötü. Ama kastımızı anlamak açısından tabii doğru mu? Evet demek yanlış mı? Onu kestiremiyorum.

Participant-1: Good morning again. Well, I have a long history with Havelsan in software engineering, software architecture, project management, scrum master roles, and architecture in general. At one point, we worked on Havelsan's software development infrastructure with Eray, and we had opportunities to use these tools extensively. We even worried about how to integrate them comprehensively. The scope of this question, of course, is not entirely clear. The levels of visualization can vary greatly. For example, with the Azure DevOps I am currently using, I use it for both work items and source control items. There are normal relational visuals, but we cannot fully understand the power and capabilities of the visualization being referred to here. So, yes, they do visualizations. You can track, navigate, and move from one item to another, following the work. But in terms of understanding what exactly is meant, is it right to say yes? Is it wrong? I can't quite determine that.

5:35- 7:16

L: Genel olarak sorumuz size ne çağrıştırıyorsa o şekilde sormayı amaçladık.

L: We intended to ask the question in whatever way it resonated with you.

Katılımcı-1: O zaman evet.

Participant-1: Then yes.

L: Başka bir yorumda bulunmak isteyen olur mu?

L: Would anyone else like to comment?

Katılımcı-2: Ben bir yorumda bulunabilir miyim. Merhaba ben (Katılımcı adı), Novelabs adlı bir şirkette çalışıyorum. Orada şu an yazılım geliştirici olarak çalışıyorum. Daha öncesinde farklı şirketlerde çalışıyordum ve bu şirketlerde çalışırken yoğun olarak bu visualization based software genel kullanımı bizde müşteri analitiklerini grafikler halinde görmek. Bunlar dışında da yazılımlarımızın loglarını incelemek, nasıl davrandığının nasıl paternlerin olduğunu görebilmek amacıyla gene grafikler aracılığıyla görmek gibi yazılımlar kullandık. Onun dışında da kendim yüksek lisans yaparken en yoğun olarak kullandığım zaman bir social network isimli bir dersimiz vardı. Orada işte şeyler arasında notlar arasındaki ilişkileri görebilmek için kullandığımız toollar vardı. Daha çok benim aklıma gelen bunlardı yani.

Participant-2: May I comment? Hello, I am (Participant's name), and I work at a company called Novelabs. I currently work there as a software developer. Previously, I worked at different companies, and during my time there, we heavily used visualization-based software primarily to see customer analytics in graphical form. Additionally, we used such software to examine logs of our applications, to see how they behaved, and to identify patterns, all through graphical means. Besides that, while I was doing my master's degree, the time I used visualization tools the most was during a course called Social Network. We used tools to see the relationships between notes. These are what mainly come to my mind.

L: Hangi amaçla kullanıyordunuz? Özellikle o dediğiniz araçları?

L: What was the purpose of using these tools, especially the ones you mentioned?

Katılımcı-2: Social network açısından kullandığımız şeyden bahsediyorsanız eğer, orada gördüğüm şey bu şeyler arasında. Bir ortamda bulunan bir şey networkte bulunan nodelar arasında bulunan ilişkileri bu paternleri çıkartabilmek, bunları görebilmek amacıyla kullanıyorduk.

Participant-2: If you are asking about the social network aspect, the main purpose was to identify and see the patterns and relationships between nodes in a network, within a certain environment.

L: Çok teşekkürler.

L: Thank you very much.

7:16-8:45

Katılımcı-4: Ben de (Katılımcı adı), şu an master öğrencisiyim. 3 yıllık bir sektör deneyimim de var. Benim kullanma şeklim yine aynı sizin gibi. Bir tane Social Network dersi almıştım. Onda da proje yapıyorduk. O projede de biz de, hani developerlar ve en azından farklı commitler arasındaki ilişkileri gösterecek bir şey yapalım diye, küçük bir denememiz olmuştu.

Participant-4: I am also (Participant's name), currently a master's student. I have 3 years of industry experience. My usage is similar to yours. I took a Social Network course. We were doing a project in that course. In that project, we also made a small attempt to show the relationships between developers and at least different commits.

L: Başka bir katkıda bulunmak isteyen olur mu?

L: Would anyone else like to contribute?

Katılımcı-3: Yani bizim de şu an kullandığımız araçlar var ama görselleştirme... Hayır mı dedim, evet mi dedim, çok da hatırlamıyorum ama büyük ihtimalle hayır deme ihtimalim yüksek. Daha çok hani text tabanlı araçlar kullanırsınız ama mesela hocamın dediği gibi, mesela logların takibinde görsel bir aracımız var aslında. Öyle düşününce aslında biz de kullanıyormuşuz gibi de düşünebilirim. Yine veritabanı tasarımları, dizaynları yaparken kullandığımız çeşitli araçlar var. Onlar da aslında bu kategoriye girerse belki onları da kullanıyoruz gibi de düşünebiliriz ama daha çok biz hani text tabanlı çalışırız gibi de düşünüyorum.

Participant-3: Well, we also have tools that we use currently, but visualization... I don't remember if I said no or yes, but I probably said no. We mostly use text-based tools, but as my colleague mentioned, we do have a visual tool for tracking logs. So, thinking about it that way, I might say we use it too. We also have various tools we use for database design and designs. If those are considered in this category, we might say we use them too, but mostly we work with text-based tools, I think.

E: Siz nerede çalışıyorsunuz hocam?

E: Where do you work?

Katılımcı-3: DIA Yazılım'da çalışıyorum. Uygulama geliştirme müdürüyüm orada.

Participant-3: I work at DIA Software. I am the Application Development Manager there.

L: Tamamdır, devam edelim.

L: Alright, let's continue.

8:45 - 9:54

L: Pre-Survey\_4. Soru: Daha sonrasında aslında yine benzer bir soru sormuştuk. Görselleştirmenin proje sürecindeki insightları yani anlamamıza katkısını sormuştuk. Genel olarak herkes zaten pozitif olarak değerlendirmiş. Bu soru üzerinden yorum yapmak isteyen olur mu?

L: Pre-Survey\_4. Question: Then, we asked a similar question about the contribution of visualization to understanding insights during the project process. Generally, everyone evaluated it positively. Would anyone like to comment on this question?

Katılımcı-1: Doğal ya, galiba evrimsel bir durum. Bizim görsel zekamız evrimsel olarak diğer duyularımızdan daha ileride.

Participant-1: It's natural, I think it's an evolutionary situation. Our visual intelligence is evolutionarily more advanced than our senses.

Katılımcı-2: Ben 4 verenlerden birisi olarak, visualization için çok kuvvetli bir araç olmasına rağmen bazen bazı abstraction orada yaptığımız bazı özetlemeler bizim bazı detayları kaçırmamıza sebep olabiliyor diye düşünüyorum. Onu onun payını vererek tam değil de biraz daha eksik puan vermiştim.

Participant-2: As one of the people who gave a 4, I think that although visualization is a very powerful tool, sometimes some abstractions and summaries we make there can cause us to miss some details. That's why I gave it a slightly lower score, considering that aspect.

Katılımcı-1: Görselleştiren çıkardığı engelden bahsediyorsun. Görselleştirici. Şey, kapsayıcılık açısından daha da kalabilir.

Participant-1: You are talking about the obstacle created by visualization. Visualizer. Well, it can stay more in terms of inclusiveness.

Katılımcı-2: Evet, aynen öyle söylüyorum.

Participant-2: Yes, that's exactly what I'm saying.

9:54 - 10:50

L: Pre-Survey\_5.Soru: Diğer sorumuzda da aslında proje süresince en fazla etkileşime geçtiğiniz software artifact'ları sormuştuk. Burada da gördüğüm kadarıyla en fazla kod dosyaları, issue'lar, commit'ler ve onu takip eden de pull request'ler seçilmiş. Bu sonuçlar üzerine konuşmak isteyen olur mu, sizce neden bu artifact'lar daha önde çıkmış?

L: Pre-Survey\_5. Question: In our other question, we asked which software artifacts you interacted with the most during the project. From what I can see here, the most frequent ones

are code files, issues, commits, and followed by pull requests. Would anyone like to discuss why these artifacts are more prominent, in your opinion?

Katılımcı-1: Valla sizin Graph model'e göre de bakınca zaten dördünü de kapsıyorsunuz. File var, issue var, commit var. Pull request zaten arada yapay bir artifact belki ondan dolayı tamamlıyorum.

Participant-1: Well, according to your Graph model, you already encompass all four. There are files, issues, commits. Pull requests are already artificial artifacts in between, so I might be completing it for that reason.

L: Geçelim o zaman ilerleyelim.

L: Let's move on then.

10:50 - 12:30

L: Pre-Survey\_6.Soru: Daha sonra, hiç projelerinizde expert developer bulunması için harici bir araç kullanıp kullanmadığınızı sormuştuk. 1 katılımcı kullandıklarını söylemiş. Nasıl bir araç kullanıyorlar? Ondan bahsedebilir mi?

L: Pre-Survey\_6.Question: Earlier, we asked whether you have ever used an external tool to find expert developers in your projects. One participant mentioned they have used such a tool. Can you elaborate on what kind of tool you use?

Katılımcı-1: Evet, evet yani tabii soruyu anlarken olabiliyor bu işte. Yani SonarQube ya da geliştirme ortamının kendi üzerinde olan araçlardan bahsediyorum ben burada. Direkt tüm değişiklik olarak almayan partiler ama her iki noktada ayrı ayrı barınabilen birisi developer'ın önünde barınabilen onlar mesela grafik yetenekleri daha farklı olabiliyor. Ayrı ayrı karşılaştırabiliyor. Ya da işte SonarQube gibi uygulamalar başka şeyler de çıkarabiliyor. Yani tümleşik olmayan yapımın haricinde. Kastettiğim buydu.

Participant-1: Yes, yes, well, of course, it depends on how you interpret the question. I was referring to tools like SonarQube or those built into the development environment. They are not directly integrated with all changes but can reside separately, like graphics capabilities that can vary. They can compare separately. Or applications like SonarQube can reveal other things outside of the integrated structure. That's what I meant.

U: Onlar doğrudan expert ya da reviewer öneriyor mu?

U: Do they directly suggest experts or reviewers?

Katılımcı-1: Aslında öneriyorlar ya, şey.

Participant-1: Actually, they do, sort of.

U: Yoksa siz analizi yapınca mı çıkarıyorsunuz? Yani doğrudan isim veriyorlar mı size, yani şu şey için mesela pull request için ya da şu file için expert gibi.

U: Or do you derive it from the analysis? Do they give you direct names, like suggest

Katılımcı-1: Ha yok yok işte pipeline analizi diye ondan dolayı söylüyorum zaten direkt pipeline'da yer almıyorlar, düz akışta yer almıyorlar, harici olarak başvuruyorsunuz, detaylandırma için başvuruyorsunuz. Yoksa şeyde yani SonarQube de aslında düz akışta direkt yer alan bir araç değil. O da kendi kendine giriş çıkış yapıyor.

Participant-1: No, no, that's why I mentioned pipeline analysis. They are not directly part of the pipeline or flow. You refer to them externally for detailing. Otherwise, SonarQube itself is not directly part of the flow either. It operates independently.

L: Devam edelim.

L: Let's continue.

12:30 - 13:22

L: Pre-Survey\_7.Soru: Son sorumuzda da aslında process smelleri ile ne kadar tanıdık olduğunuzu, ne kadar bildiğinizi sormuştuk. Katılımcıların çoğu zaten bu konu hakkında bilgisi varmış aynı zamanda. Çalışma ortamınızda daha önce hiç process anomalilerini tespit eden bir uygulama kullanmış mıydınız diye sormuştuk. Fakat sanırım kullanan olmamış.

L: Pre-Survey\_Question 7: In our last question, we asked about your familiarity with process smells and how much you know about them. Most participants seem knowledgeable on this topic. We also asked if you have ever used an application to detect process anomalies in your work environment, but it seems no one has.

Katılımcı-1: Öyle bir uygulama var mı?

Participant-1: Is there such an application?

E: Var hocam.

E: Yes, there is.

E: Son katılımcı siz de kendinizi tanıtır mısınız?

E: Could the last participant introduce themselves?

Katılımcı 5: Ben de Dia yazılımda çalışıyorum, katılımcı 3 ile birlikte orada software developer olarak çalışıyorum.

Participant-5: I work at Dia Software as a software developer alongside Participant 3.



14:34 - 17:50

Katılımcı 1: Yani biz kullanıcı olarak ben bunu nasıl kullanacağım, hangi değeri alacağım? Bu araç ile analiz yapma motivasyonumuz ne, yani?

Participant-1: As a user, how would I use this? What value would I get? What is the motivation to perform analysis with this tool?

L: Karar alma durumlarında destek sağlamak (Açıklama Çalışma Özetlenir)

L: To support decision-making situations (Explanation of Study Summarized)

Katılımcı 1: Geliştiricinin mi yoksa yöneticinin mi?

Participant-1: For developers or managers?

U: Geliştiricinin de yöneticinin de yani. Yönetici de mesela şöyle soruları yanıtlayabilir işte. Herhangi bir file ya da file grubu hakkında ekspert aradığı zaman iyi bir expert bulamazsa işte burada eksiğimiz var diyebilir ya da o konuda sadece bir expert bulursa ya bu adam giderse ne yaparız diyebilir. Yani öyle şeyler yapabilir ama daha da önemlisi mevcut sistemde var olan problemleri dikkat etmek, yani anormallikleri tespit etmek, birileri tutarlı olarak işte kendi açtığı issue'ları. Kendi kapatıyorsa bu hoş bir şey değil. Yani bunları yakalamak örneğin. Yani bu sadece bir örnek, yani developer process sırasında alınan kararlara destek vermek en önemlisi yani.

U: For both developers and managers. A manager, for example, can answer questions like if they are looking for an expert on any file or group of files and cannot find a good one, they might identify it as a gap. Or if they find only one expert, they might think, "What will we do if this person leaves?" More importantly, it's about identifying existing problems in the system, like detecting anomalies. For instance, if someone consistently opens and closes their own issues, that's not a good practice. It's about catching such things. This is just one example. The most important thing is to support decision-making during the developer process.

Katılımcı 3: Kodun kalitesi mesela bununla benzer bir ölçüm yapılıyor mu?

E: (Çalışmanın scope'u anlatılır)

20:10 - 29:55 ( Presentation Session About the Concepts)

30:00 - 47:27 (BREAK)

47:27 - 01:08:56 ( SAA DEMO)

51:06 - 52:10

Katılımcı 1: Bu gruplara bir dış etiket güzel olabilir.

Participant 1: A label on these groups would be nice.

L: Nasıl bir etiket hocam?

L: What kind of label?

Katılımcı 1: Ya biraz uzaklaşınca levhası mesela göremiyoruz ya tam gözüküyor, daha büyük, developer'ın ismini yazma. Grup üzerine bir etiket güzel olabilir.

Participant 1: You know, when you move away a bit, you can't see the sign clearly. Writing the developer's name bigger on the group would be nice.

Katılımcı 1: Bu mesela collapse ettiğiniz zaman ilişkiler çok da gözüküyor ya o da algılamayı zorlaştırıyor. Çoklu ilişkiye gerek yok, tekli olması yeterli.

Participant 1: When you collapse it, the relationships become less visible, making it harder to understand. Single relationships are sufficient; multiple relationships are not necessary.

53:11 - 1:00:28

Katılımcı 1: E şu an operasyonel seviyede değiliz zaten. Yani geçmiş veri için kullanım amacımız nedir bunda?

Participant 1: We are not at an operational level right now. So, what is our purpose for using historical data in this?

L: Hocam onu da şöyle düşünebiliriz mesela siz günde bir kez SAA verisini güncelleyip aslında operasyonel seviyede tutabilirsiniz.

L: You can update the SAA data once a day to keep it at an operational level.

U: Operasyonel ortam olarak düşünebilirsiniz. Bunun düzenli olarak güncellendiğini farz ederim.

U: You can think of it as an operational environment. Assume it is regularly updated.

Katılımcı 1: Yani öyle baktığımız zaman bunun doğal yeri aslında şey işlemin gerçekleştirildiği yerdir ya o daha pull request yöneten yer neresi git. Git'in üzerinde falan olması lazım. Yani bu git'e eklenti olması lazım. Bu şeyin özelliği git'te yer alan bir şey olması gerekebilir. Oraya ajan koymak gerek, ters ajan.

Participant 1: When we look at it that way, the natural place for this is where the process is carried out, like where pull requests are managed, which is Git. It should be on Git. It needs to be a feature within Git. You would need an agent there, a reverse agent.

E: Orada da var zaten.

E: It's already there.

Katılımcı 1: Şimdi düşünüyorum ki zaten biz var olan veriler arasında geriye dönük bir çalışma yapıyoruz. Şu anki şey yaklaşık regresif bir yaklaşım. Doğal olarak şu ana kadar bulunduğumuz yer operasyonel değil. Bu pull request açık olabilir veya kapanmış gitmiş de olabilir. Yani işlemi tamamlanmış da olabilir. Burada reviewere önermemizin değeri nedir?

Participant 1: We are currently working with existing data retrospectively. Our current approach is approximately regressive. Naturally, we are not at an operational level yet. This pull request may be open or closed and completed. So, what is the value of recommending a reviewer here?

E: Tamam, iki şey var. Burada incremental olarak düşünelim.

E: Okay, let's think incrementally.

Katılımcı 1: Orada da şöyle düşün hocam. Yani ben GitHub kullanıyorum. Bu da ayrı bir araç gibi düşünüyorum. Yani GitHub'ın içerisinde gömüldüyse problem güzel çözülmüş oluyor ama ayrıysa iki araç kullanmayı öneriyorsun demektir.

Participant 1: Think of it this way. I use GitHub, and I see this as a separate tool. If it's embedded within GitHub, the problem is nicely solved, but if it's separate, you're suggesting using two tools.

E: Yok yok bizim zaten GitHub botumuz da var.

E: No, no, we also have a GitHub bot.

Katılımcı 1: GitHub'daki pull request'in içine bunu konumlandırman gerekecek. Ayrı bir hücre olarak. Bir eklenti gibi şuradaki kutuyu oraya koymamız gerekecek işte.

Participant 1: You would need to place this within the GitHub pull request, as a separate cell. Like an add-on, we would need to place the box there.

E: Tamam, bizim GitHub'u konfigüre ettiğimiz zaman şey yapabilir zaten pull request açıldığı zaman arkada bunu çalıştırır. Buraya linkli buraya da gelebilir. Yani öyle diyorsun değil mi?

E: Okay, when we configure GitHub, it can already run this in the background when a pull request is opened. It can link here as well. That's what you mean, right?

Katılımcı 1: O zaman orada olmaz. Operasyonda olması lazım.

Participant 1: Then it should be in operations.

Katılımcı 4: Anladığım kadarıyla GitHub'da da reviewer suggestion ederken mesela o file'dan önceden çalışanları yani örnek veriyor. Mesela şunlar recommended reviewer diyor galiba. Siz öyle bir şey arıyorsunuz. Hani direkt GitHub'ın kendi sayfasında verebilirim, yönlendirmek yerine diyorlar.

Participant 4: As far as I understand, GitHub also recommends reviewers, for instance, suggesting those who have previously worked on that file. They say something like these are recommended reviewers. You are looking for something like that, to provide directly on GitHub's own page instead of redirecting.

Katılımcı 1: O zaman tam senin dediğin sana o zaman.

E: İki taraf da var yani GitHub'dan SAA'ya gelmek de var, SAA'dan GitHub'a gitmek de.

E: There are two sides to this, both coming from GitHub to SAA and going from SAA to GitHub.

Katılımcı 1: Değerlendirmesinin sonucunu direkt GitHub yani reviewere önermesi ile beraber operasyonel seviye eğitimden geçecek. Çizim kanvasında konması gerekecek.

Participant 1: The outcome of the assessment should be directly suggested to the reviewer in GitHub, passing through the operational level. It should be placed on the drawing canvas.

L: Hocam onu da şu şekilde yapabiliyoruz. Fotoğrafını koyabiliyoruz.

L: We can do that by adding a photo.

Katılımcı 1: Bak şöyle düşün, ben şimdi geliştiriciyim ya da mesela geliştirme yöneticisiyim. Pull request açıldı. Tamam pull request açıldıktan sonra kimin konması gerekiyor. Hatta onun otomatik hale getirmek gerekir. Önerilerini birinci koy yani değiştirilmezse devam eder diyebiliriz. Üretici tarafı için. Ama operasyonda seviye işledi. Kural ihlaline bakmıyoruz, kural koyuyor muyuz bilmiyorum.

Participant 1: Think of it this way, I am a developer or a development manager. A pull request is opened. After the pull request is opened, it needs to be automatically suggested who should be assigned to it. We can say that if the suggestions are not changed, they continue as is. That's the production side of things. But at the operational level, we need to see if a rule violation occurred. I don't know if we are setting rules.

E: O process analiz var da bu dediğini yani güzel bunu kolaylıkla yapabiliriz diyorum ya yani orada botumuz var zaten, botumuz kendi kendine trigger olacak.

E: There is process analysis, and we can do this easily because we already have a bot that triggers itself.

U: İnsanlar n tool kullanıyorsa mesela n+1 tool elbette istemez ama n+1 toolun içinde n toolun yapamadığı bir şey varsa ne yapacak, n+1 kullanmam mı diyeceksin yani? Öyle de düşünebiliriz.

U: If people are using n tools, they certainly wouldn't want n+1 tools unless the n+1 tool can do something that the n tools can't. In that case, wouldn't they use n+1?

Katılımcı 1: Ben hayır derim genelde.

Participant 1: I would generally say no.

U: Kaç tane tool kullanıyorsunuz siz mesela?

U: How many tools do you use, for example?

Katılımcı 1: Normalde en az seviyede olmasını istiyorum. Bir veya ikinin dışına pek fazla çıkmak istemiyorum çünkü zaten yönetemiyorum, kafa olarak da yönetemiyorum.

Participant 1: I usually want to keep it to a minimum. I don't want to go beyond one or two because I can't manage it, mentally as well.

E: Tamam olay şu, varsa orada bizim GitHub botumuz direkt bunu run edip orada gösterseydi bu çözüldü değil mi? Tamam okey bu zaten yapılabilir bir durum, bu güzel bence.

E: Okay, so if our GitHub bot runs this directly and shows it there, the problem is solved, right? This is already doable, and I think it's good.

Katılımcı 1: Ben başaramıyorum yani birden fazla tool kullanmayı.

Participant 1: I can't manage using more than one tool.

Katılımcı 4: Ben junior developer olarak ilk geldiğimde reviewer aramak için yeteri kadar şey olmuşum. Artık öyle bir çaresizliğe girmiştik ki. Başka bir tool'u kullanmak mesela beni o kadar rahatsız etmezdi.

Participant 4: When I first started as a junior developer, I didn't have enough experience to find a reviewer. I was so desperate that using another tool wouldn't have bothered me that much.

Katılımcı 1: Ayrı bir konuda, geçmiş verilere dayandığı için var olana tapınmaya neden olabilir. Geçmiş bugün de kendini tekrar ettiği zaman mutlaka ikinci kişinin olması lazım. Daha az kalifiye olan falan şimdi o manuel operasyona ya da rotasyona girmesi lazım gibi.

Participant 1: On a separate note, relying on historical data can cause us to idolize the existing data. When the past repeats itself today, there must be a second person involved. Less qualified people should manually or rotationally handle this.

U: Visualization getirdiği başka avantajlar var yani? Normalde sadece bir tool'un önerdiği skora bakmaksızın visualization sayesinde kararınızı da değiştiriyorsunuz. Biraz daha devam edersek göreceğiz.

U: Visualization brings other advantages. Normally, without looking at the score suggested by just one tool, visualization can also change your decision. We will see more if we continue a bit.

1:01:00 - 1:05:20

Katılımcı 1: Skor hesabı da çok desteklenen bir konu değil aslında.

Participant 1: Score calculation is not a highly supported topic, actually.

Katılımcı 3: Bu skor nasıl hesaplanıyor? Mesela biz 3.34'ü nasıl anlamalıyız?

Participant 3: How is this score calculated? For example, how should we interpret a score of 3.34?

L: Şimdi aslında çok detaya girmeden anlatmak gerekirse, developer ve değişen kod dosyası seti arasındaki ilişkiler üzerinden giderek hesaplanıyor.

L: Well, to explain without going into too much detail, it's calculated based on the relationships between the developer and the set of changed code files.

Katılımcı 2: Mesela biz şimdi şey mi anlamalıyız? Dev-1 daha çok çalışmış, Dev-2 daha az.

Participant 2: So, should we understand it as Dev-1 has worked more, Dev-2 less?

Katılımcı 1: Buradaki skor daha önceki değişen kod setine bulaşıklığını gösteriyor. Mesela o dosyalara bulaşıklığı gösteriyoruz.

Participant 1: The score here shows the involvement in the previous set of changed code. For example, it shows their involvement with those files.

E: Mesela E.java developer-1 daha fazla bulaşmış, daha fazla review etmiş.

E: For instance, E.java shows that developer-1 has been more involved and has reviewed it more.

Katılımcı 3: Yani bu dosyayla kim daha çok ilgilendi onun skorlanması, ben review atarken benim işimi kolaylaştırabilir.

Participant 3: So, scoring who has been more involved with this file can make my job easier when assigning reviews.

Katılımcı 2: Bir dosya birden fazla olunca ne oluyor?

Participant 2: What happens when there are multiple files?

L: Aynı şekilde bütün dosyalar bazında bir skor hesaplanıyor. Sonra topluyoruz. O dosya setini toplam en iyi bilen kişiyi önermeye çalışıyoruz.

L: We calculate a score for all files in the same way. Then we sum them up. We try to recommend the person who knows the entire set of files the best.

Katılımcı 1: Buna mesela şey aklıma geldiği için paylaşayım belki değer yaratılır diye. Yani şöyle bir düşünce, bilmem ne yenileniyor mu? Aslında bir şeyler tutuyorlar, sürekli kontrolü tabi de araştı tutuyor ama. Denormalize olarak böyle dosyaların içerisinde en çok değiştiren ya da en çok review eden bilgisi yer almıyor. Aslında denormalize bilgi güzel, bilgi bak aradığını direkt oradan da bulabiliyorsun ya normalde burada en çok kim yazmış, en çok kimi gözden geçirmiş deseydi diye değil mi? O da oradan direkt denormalize bilgi olarak geri alırsa bunların da geri dönüşü hep kolay olabiliyor.

Participant 1: This reminds me of something that might add value. There's a thought, is anything being renewed? They keep some things continuously monitored in the tool. However, the information about who has changed or reviewed the most isn't always denormalized in the files. Actually, denormalized information is good because you can directly find what you're looking for. Normally, you would ask who has written the most, who has reviewed the most, right? If it gets this information back as denormalized data, it can make things easier.

E: Tamam, tam olarak bunu yapıyor.

E: Exactly, that's what it does.

Katılımcı 1: Aynı zamanda şey için de gerekiyor, düzenleyici için de gerekiyor, yani bana bir bug geldi kime göndereceğiz bunu ha tabi. Bunu denormalize bilgi olarak tutmak lazım, source code'un yanına yazmak lazım.

Participant 1: It's also needed for the organizer. For example, if I get a bug, to whom should I send it? It's necessary to keep this as denormalized information, written alongside the source code.

1:09:00 - 1:22:52 (SAA Framework Anlatımı) (Yemek Yendi)

1:23:00 - 2:01:00 (User Trial Session)

2:01:45 - 2:03:40

Katılımcı 1: Şey yok değil mi, böyle Dashboard düşüncesi var mı? Mesela bir insight oradan geliyor. Normalde başlangıçta bir görüşe sahip olamıyorum. Eğer özel olarak polislik yapmak istediğim bir konu yoksa dashboard bir içgörü oluşturur.

Participant 1: There isn't anything like this, right? Is there any idea of a dashboard? For example, getting an insight from there. Normally, I can't have an opinion at the beginning. If there's not a specific issue I want to focus on, a dashboard can provide an insight.

E: Doğru, evet. Güzel.

E: Correct, yes. Nice.

Katılımcı 1: Sana şey gönderirim.

Participant 1: I'll send it to you.

L: Herkes zaten yanıtı var, tamam.

L: Everyone already has the answer, okay.

E: Göndermiş mi? Tamam.

E: Has it been sent? Okay.

L: Tamam o zaman devam edelim.

L: Okay, then let's continue.

L: İlk önce birkaç tane genel sorumuz var, daha sonrasında aslında bu yaptığımız task'lar hakkında birkaç soru soracağız.

L: First, we have a few general questions. After that, we'll ask some questions about the tasks we performed.

2:03:40 - 2:11:08

L: İlk sorumuz şu şekilde: aslında size sunumda yaptığımız bu SAA'nin graph modelini ne kadar yeterli buldunuz? Projeyi yansıtmak açısından önümüzdeki sayfalardan inceleyebilirsiniz modeli. Bu bazda alt sorumuz, siz olsaydınız hangi tür software artifact'lerin, ilişkilerin veya özelliklerin eklenmesi gerektiğini düşünürdünüz? Sizce modele neler eklenebilir ekstra olarak bunu sormuştur.

L: Our first question is: how adequate did you find the graph model of the SAA presented to you during the presentation? You can review the model in the upcoming pages to reflect on the project. Based on this, our sub-question is, if you were in charge, what type of software artifacts, relationships, or features would you think should be added? What additional elements do you think could be included in the model?

U: Bu discussion açısından yani şey değil, herhangi bir doldurmuyoruz yani. Özetle diyoruz ki mesela sizin aklınızda bir use case varsa o use case için gerekli olabilecek yeni bir artifact. Farklı bir artifact ya da farklı bir relation var mı?

U: This is for discussion purposes, so we are not filling out anything specific. In summary, we are asking if you have any use cases in mind, are there any new artifacts or different relations that could be necessary for that use case?

Katılımcı 2: Ben genel olarak şeyi gördüm. Issue tipleri ile ilgili bir şeyler olabilir. Yani genel olarak bug'lar üzerinden gitmiş herhalde işte feature'lar feature'ların geliştirilmesi ve bu feature'lardan dolayı oluşan bug'lar arasındaki ilişkiler ile ilgili bir şeyler yapılırsa bence şey için çok kullanışlı olur. Yani biraz hani yaptığımız şey şu şekilde, bugün bir feature çıkıyoruz hemen



bundan bir gün sonra bir anda bug sayımız 5 tane 10 tane artıyorsa. Burada bir şey olabilir, onların da burada represent edilmesi güzel olabilir. Aklıma gelen ilk şey bu oldu. Burada yaparken bu eklenir dedim.

Participant 2: In general, I noticed something related to issue types. It mostly seemed to focus on bugs. If something could be done about the relationships between features, the development of features, and the bugs caused by these features, I think it would be very useful. For example, if we release a feature today and then suddenly our bug count increases by 5 or 10 the next day, there could be an issue. It would be nice to represent that here as well. That's the first thing that came to my mind. I thought this could be added while we were doing this.

U: Issue typeları baştan biz bir şekilde entegre edelim dedik ama galiba Jira'dan gelmiyordu değil mi?

U: We initially wanted to integrate issue types somehow, but I think it wasn't coming from Jira, right?

L: Şu an type şeklinde bir property olarak tutuyoruz.

L: Currently, we are keeping it as a property called type.

Katılımcı 1: Type'ların bir hiyerarşisi var. Bir feature, daha büyük bir şey task aslında onun altında çalışılmasını sağlayan şey bildiğiniz.

Participant 1: There is a hierarchy of types. A feature is a larger task, and under that, there are smaller tasks that contribute to it.

U: Ben hiç Jira'dan yok diye biliyordum da onun için var yani.

U: I thought it wasn't available from Jira, but it is.

Katılımcı 2: Property olarak da kullanılmış zaten anladığım kadarıyla fakat belki de tamamen farklı bir artifact olarak düşünülmesi ilişkileri bulmak açısından iyi bir şey olabilir diye düşünüyorum.

Participant 2: As I understand it, it is already used as a property, but maybe thinking of it as a completely different artifact could be good for finding relationships.

Katılımcı 1: Biraz daha özelliğe doğru kaymak lazım. Yeni özelliklerin neden olduğu hatalar gibi.

Participant 1: We should move towards more features, like the bugs caused by new features.

E: Ya bu process smeller daha spesifik olduğu için ona girmemiştik ama doğru yani bug feature task işte epic falan. Hani bunlar kesinlikle gelebilir.

E: We didn't include those because process smells are more specific, but you're right. Bugs, features, tasks, epics—they could definitely be included.

U: Yani issue type specific use caseler.

U: So, issue type specific use cases.

Katılımcı 2: Evet, evet.

Participant 2: Yes, yes.

Katılımcı 1: Aslında şöyle başlıyor, ya olay işleyen teslim etmiş işe başlıyoruz, iş değeri var orada işlerini sağlayan şeyde yani kültür aslında mimarlık kavram da yani kabiliyet ya da şey user story diyelim. User story işlerini sağlayan işte User story başka bir bakış açısı.

Participant 1: It actually starts with delivering a working feature. There is a job value there that supports the work, meaning the culture or architecture concept—let's say capability or user story. A user story supports work from a different perspective.

Katılımcı 1: Ya user story ile başlayıp sahada teslimat sağlayan aslında bir de şey var build var. Gerçekte execute bloğu. En solda bir inisiyatif var, iş inisiyatifi en sağda da çalışan yani deployment girmek belki gerekmez ama build var.

Participant 1: Starting with a user story and providing delivery on the field, there is also the build aspect. In reality, it's the execution block. On the far left, there is an initiative, a job initiative, and on the far right, the actual working deployment, maybe not necessary to include, but there is the build.

Katılımcı 1: Yani sol ve sağ taraflarında 2 eksikimiz var. Sol tarafta hocam söylediler zaten sağ tarafta da bana göre build eksik.

Participant 1: So, we have two gaps on the left and right sides. On the left side, as mentioned, and on the right side, I think the build is missing.

L: Peki hocam mesela şeyi de sormak istiyordum, bunda ekleseniz mesela ekleyebilir miyiz?

L: Well, for example, could we add that if we wanted to?

Katılımcı 1: Olur tabi ya eklenir.

Participant 1: Of course, it can be added.

U: Yani senin söylemek istediğin şu herhalde mesela bu yeni eklenecek build artifact'i ve diğerleri arasındaki ilişkileri tutan commonly used araçlar var mı? Yani mesela Jira bir sistemiyle entegre mi?

U: So, what you're trying to say is, are there any commonly used tools that track the relationships between this new build artifact and the others? I mean, for example, is it integrated with a system like Jira?

Katılımcı 2: Tam olarak var zaten şey gibi de düşünebilirsiniz. Jira'nın zaten GitHub yerine kullanabileceğiniz Bitbucket diye de şeyi var. Direkt şey yaptınız. Bana zaten öyle oluyor. Onunla da entegrasyonları var.

Participant 2: Exactly, you can think of it like that. Jira already has Bitbucket that you can use instead of GitHub. You did it directly. It happens to me like that. There are integrations with that as well.

Katılımcı 2: Şeyde de şöyle bir durum var, commitler yerine deployment'lar kullanılsa daha doğru bilgiler verebilir diye düşünüyorum. Çünkü her commit bir şey ifade etmeyebilir ama her deployment product için bir değişiklik demektir. Gerek farklı bir çözüm gerekse yeni bir feature ekleme.

Participant 2: There's also this situation, I think it would provide more accurate information if deployments were used instead of commits. Because not every commit means something, but every deployment means a change for the product. Whether it's a different solution or a new feature addition.

Katılımcı 1: Bakıyorum da, committe gerekiyor, dosyaları olmuyor.

Participant 1: I see that, but commits are necessary, files don't work otherwise.

Katılımcı 2: Tabi gerekiyor ama ben daha doğru bir bilgi olduğunda.

Participant 2: Of course, they are necessary, but I'm talking about more accurate information.

Katılımcı 1: Çünkü değer, ürün çalışan ürün tarafından bizim söylediğimiz.

Participant 1: Because the value comes from the working product, as we said.

U: Bir tanesi internal testinde yakaladığım buglar, bir tanesi de canlıda yakaladığım buglar aslında. Ya da sorunlar.

U: One is bugs you catch in internal testing, and the other is bugs you catch in live environments. Or issues.

Katılımcı 1: Hocam aslında temel ilkemiz şu, yani biz geliştirme yapmak için geliştirme yapmıyoruz. Temel derdimiz bug bulmak kod yazmak değil, temel derdimiz normal çalışan bir ürün ortaya çıkarmak, çalışan ürün temel dizisi bir iş hikayesi çıktısı da hakikaten işte executable. Ortadaki faaliyet yapay, bizim geliştirme mühendisinin kendi sorunları yani. İşin kendisinin buna ilgili bir şeyi yok, kaygısı yok. Bug bulduk kod yazdık commit ettik falan onlar bizim sorularımız aslında işle ilgili bir sorun değil, işin sorunu benim ihtiyacım bu çalışan ürün nerede. Board'daki bölüm bizim software development'ın kendi problemleri. Ya iş dünyasının onunla ilgisi bir karşılığı yok.

Participant 1: Actually, our main principle is this, we don't develop just for the sake of development. Our main concern is not finding bugs or writing code, our main concern is to create a working product, and the main sequence of the working product is indeed an executable output of a business story. The activity in the middle is artificial, it's our development engineer's own problems. The business itself has no concern or relation to this. We found bugs, wrote code, committed them, etc., those are our issues, not a business issue. The business issue is where is my needed working product. The section on the board is our software development's own problems. The business world has no counterpart to it.

E: Ama aynı zamanda bizim artifact'leri mine ederek bir şey çıkarabiliriz.

E: But at the same time, we can extract something by mining our artifacts.

Katılımcı 1: Sponsorumuz ne? User story. Evet tamam o gerekiyor çünkü o olmadan boş çalışıyoruz ya o olmazsa hiçbir şey yapmıyoruz anlamında kendi kendimize bir şey yapmamız mümkün değil, keza onun da teslim etmesi gereken en sonunda çalışan bir kod. Yani bu formüle kadar gidebilir build'e.

Participant 1: What is our sponsor? User story. Yes, it is necessary because without it we are working aimlessly, we can't do anything without it, we can't do anything on our own, and it also has to deliver a working code in the end. It can go as far as the build.

E: Tamam user story, build, artifact'leri eklenmeli. Onları yazdık.

E: Okay, user story, build, artifacts should be added. We wrote them.

Katılımcı 1: Pipeline bu zaten.

Participant 1: This is the pipeline already.

E: Ondan başka bir şey var mı?

E: Is there anything else?

L: Genel olarak şu an hali hazırda olan model hakkında düşüncelerinizi de paylaşırsanız çok seviniriz.

L: In general, we would be very happy if you could share your thoughts on the current model.

Katılımcı 4: Evet çok yani değiştirilebilecek bir şey değil de şöyle bir farklılık olabilir. Burada sadece developer gibi düşünmüştük. Çünkü hani projede da zaten sadece developer oluyor, genellikle issue açan da, report eden de, solve eden de. Ama hani daha özel, hani şirket yapısında hani teknik olmayan üyeler de bir issue açabiliyor, comment atabiliyor, belki developer'a öyle bir özellik eklenebilir. Hani teknik developer mı değil mi? Gibi bir şey oluyor ama open source için çok farklı olacağını düşünüyorum ama hani şirket ortamında oluyor.

Participant 4: Yes, it's not something that can be changed much, but there could be a difference. Here, we only thought of it as a developer. Because usually in a project, the developer is the one who opens, reports, and solves issues. But in a company structure, non-technical members can also open issues, leave comments, and maybe such a feature can be added to the developer. Like whether they are a technical developer or not. But I don't think it would be very different for open source, but it happens in a company environment.

E: Doğru, güzel, tamam.

E: Correct, nice, okay.

Katılımcı 4: Mesela teknik olmayanları bilerek bazı query'lerde dışarıda bırakırız. Ya da belki hani sadece ben teknik olmayan üyeyi bulmak istiyorum. Bu konu hakkında konuşmam gereken sadece onu bulmak için.

Participant 4: For example, we can deliberately exclude non-technical members in some queries. Or maybe I just want to find the non-technical member. This is the person I need to talk to about this topic.

L: Teşekkürler, devam edelim.

L: Thank you, let's continue.

2:11:08 - 2:13:38

L: Diğer bir sorumuzda da aslında şunu sormuştuk, sizce bu software artifact analyzer tool'una yeni bir analiz eklenecek olsa ne gibi analizler eklenebilir? Sormaya çalıştığımız şey aslında şu, dediğimiz gibi biz en temelinde bu software artifact traceability graph yapısını kullanıyoruz. Sizin aklınıza mesela bu graph yapısını kullanarak aslında aynı şekilde bu sisteme entegre edilebilecek bir analiz çeşidi geliyor mu? Veya hani sizin hani çalışma süresince keşke böyle bir araç olsaydı dediğiniz bir noktada olabilir.

L: In another question, we actually asked this: If a new analysis were to be added to this software artifact analyzer tool, what kind of analyses could be added? What we're trying to ask is this: As we said, at its core, we use this software artifact traceability graph structure. Do you have any ideas for an analysis type that could be integrated into this system using this graph structure? Or is there a point where you thought, "I wish there was such a tool" during your work process?

U: Şunları da yapsaydı iyi olur 2 tanesini söylediniz zaten.

U: It would be good if it also did these things; you already mentioned two of them.

Katılımcı 1: Birinci bölümdeki zaten 2 başlık sola ve sağa doğru geliştirdim burayı yani. Once şey tarafı da var, issue'lar ile ilgili user story eklenmesi gibi, orasını da genişler. Sağda galiba.

Gelince oradan çıkan hatalara dönük bir yoğunlaşmalar da çıkar. Bu şekilde user story ile hatalar arasındaki ilişki içsellik de keşfedilmeye başlar.

Participant 1: In the first section, I developed the two headings to the left and right here. Once there's an issue side, like adding a user story related to issues, that part also expands. I think it's on the right. Focusing on the errors that come from there is also revealed. This way, the relationship between the user story and the errors starts to be discovered.

L: Başka yorum yapmak isteyen olur mu?

Katılımcı 3: Yani bana aslında bakınca şöyle hizmetler sunsa çok güzel bulurdum. Ben mesela gün içerisinde yakaladığı sıkıntıları böyle bir dashboard olur ya da herhangi bir şekilde.

Participant 3: When I look at it, it would be great if it provided such services. For example, if there was a dashboard or any way to show the issues I encountered during the day.

U: Notify etse diyorsunuz.

Katılımcı 3: Aynen, yani ben işte bugün şu issue'da sıkıntı yaşadım ya da şu commit'te bir sıkıntı yaşadım gibi gösterse ben ona göre aksiyon alsam daha güzel olur. Hocam dediği gibi şimdi çok detay var, bu detay da genelde böyle bir sıkıntılı bir durum olduğunda biz buraya gideriz. Gerçekten bu problem yaşamışsındır, kim yapmış, neden yapmış, ne olmuş, biz bilemiyoruz. Issue'da ne gibi sıkıntılar olmuş ya da belki ters taraftan da bakabiliriz. Yazılımcı tarafından da bakabiliriz. Yani yazılımcıyı daha iyi analiz edebilmek için ne sıkıntılar yaşamış, bunlar nasıl çözeriz bundan anlamak için o taraftan da bakıp sonuçlar elde edebiliriz. Bu tarz bir öneri mekanizmaları güzel olurdu açıkçası.

Participant 3: Exactly, it would be better if it showed me, for example, I had a problem with this issue today or this commit, and I could take action accordingly. As my colleague said, there are many details, and we usually go here when there's a problematic situation. We don't really know what problem occurred, who did it, why it happened, what happened. What issues were there in the issue, or maybe we can look at it from the other side. We can look from the developer's side as well. To better analyze the developer, to understand what problems they experienced, how to solve them, we can look from that side and get results. Such recommendation mechanisms would honestly be nice.

E: Tamam, teşekkürler.

E: Okay, thank you.

2:13:38 - 2:19:51

L: Diğer bir soruda aslında, bu gösterdiğimiz bu çizim kanvası projedeki software artifact'leri hakkında bilgi edinme veya genel olarak bu tarz işler için ne kadar katkı yaratıyor? Projedeki artifact'leri incelemek için bu çizim kanvasını kullanır mıydınız?

L: In another question, actually, how much does this drawing canvas we showed contribute to gaining information about the software artifacts in the project or for such tasks in general? Would you use this drawing canvas to examine the artifacts in the project?

Katılımcı 1: Güzel yaygın olarak kullanılan bir kanvas ama şeyler (node ikonları) biraz fazla büyük, kullanıcı olarak söylüyorum. Belki olayı biraz daha estetik hale getirebilirsiniz.

Participant 1: It's a nice, commonly used canvas, but the icons (node icons) are a bit too large, I'm saying this as a user. Maybe you can make it a bit more aesthetic.

U: Neler fazla büyük?

U: What is too large?

Katılımcı 1: Node ikonlar falan fazla büyük. Yani algıyı kendi üzerine çekmeye çalışıyor. İkonlar bağlamdan ziyade öne çıkıyor. Ama o şey sadece UI problemi yani.

Participant 1: The node icons and such are too big. They try to draw attention to themselves. The icons stand out more than the context. But that's just a UI problem.

Katılımcı 3: Ya bir noktada ekranda çok nesne var gibi geliyor bana. Hani belki bir şeyleri, az önce de yaptık ya, mesela bir şeye tıkladık bir şey açıldı, bir şeye tıkladım bir şey açıldı falan orada kaybolabilirim ben. Peki nasıl kaybolmamalıyım?

Participant 3: At some point, it feels like there are too many objects on the screen. For example, we clicked on something, something opened, we clicked on another thing, something else opened, and I might get lost there. So how should I not get lost?

U: Onun da yolu aslında istediğiniz bölümü seçip direkt delete ya da hide edebilirsiniz. Göstersene o bölümü.

U: Actually, you can select the section you want and directly delete or hide it. Show that section.

Katılımcı 1: Orada şeyde kullanılabilir, çözünürlüğü otomatik olarak ayarlama yani expand collapse otomatik olarak gerçekleştirme hikayelerde de var.

Participant 1: It can be used there as well, like automatically adjusting the resolution, performing expand and collapse automatically. It's also present in stories.

L: (Delete hide özellikleri çizim kanvasında gösterildi)

L: (Delete hide features were shown on the drawing canvas)

Katılımcı3: Yani photoshop uygulamalarda özel bir layer mantığı var. Mesela seçili kısmı bir layer'a aktar gibi, o layer'ı hide edince kaybolsun.

**Participant 3:** There is a special layer concept in Photoshop applications. For example, transferring the selected part to a layer, and when you hide that layer, it disappears.

L: Onda da şu var aslında, burada History bölümüne gidip eski çizimleri burada saklayabiliyoruz.

L: Actually, we can do that by going to the History section and keeping the old drawings there.

U: Şey de yapabiliyoruz, direkt bulunan graph çizimini kaydedip sonrasında yükleyebiliyoruz.

U: We can also save the current graph drawing directly and load it later.

Katılımcı 1: Elle node ekleyebiliyor muyuz? Araya gidip.

**Participant 1:** Can we manually add nodes? By going in between.

U: Yok.

U: No.

Katılımcı 1: Ben o bazen bir query'ler dışında yani ilgi çekici olabilir ama aslında işler yapılır ama şu dosyada ya da şu commit'te şu developer'ın ilgisi de var mı acaba diye?

**Participant 1:** Sometimes it's interesting to see if there is involvement from a developer in a specific file or commit beyond just running queries.

Katılımcı 2: Soruyla ilgili bir şey söyleyebilir miyim acaba? Şeyde ben bu, visual representation ve burada yapılan şeylerin şirketler için şöyle bir yararı olduğunu düşünüyorum. Biz her zaman insanların performanslarıyla ilgili veya yaptıkları şeylerle ilgili ölçülebilir şeyler bulmaya çalışıyoruz. Bunun bu şekilde history'sinin sağlanıp gittikçe gelişen, atıyorum burada Hans gösteriyor. Hans'ın şu anda bu kadar commit'te, bu kadar bug'da bir sıkıntısı vardı fakat bir sene sonra bu azaldı veya onunla bir iyileşme yaşandı şeklindeki şeyleri böyle bir ölçülebilir bir şeye koymanın gerçekten çok büyük bir değeri olduğunu düşünüyorum. Bizim için bu da her zaman eksikliğini çekilen bir şeydir diye.

**Participant 2:** May I say something about the question? I think the visual representation and the things done here have a benefit for companies. We always try to find measurable things about people's performances or what they do. Providing such a history and showing progress over time, for instance, showing that Hans had issues with certain commits and bugs, but a year later, he improved. Putting such things into measurable terms is really valuable. This is something we always lack.

U: Zaman içindeki değişimi mi diyorsun?

U: Are you talking about changes over time?

Katılımcı 2: Evet.



**Participant 2:** Yes.

U: Aslında bizim bir şey time bar'ımız var, bunu da gösteririm.

**U:** Actually, we have a time bar that shows this, I will show it to you.

Katılımcı 2: Bunun görsel olarak görülmesi bile bir kanıt oluşturduğu için gerçekten çok önemli. Genelde bunlar hissiyata göre yapılıyor veya işte en iyi yapan kapattığı issue sayısına göre yapılmış bir şey görüyorsan falan ama burada çok daha ayrıntılı ve güzel şeylerini görebiliyor olması bence çok iyi bir değer.

**Participant 2:** Seeing this visually is really important as it provides evidence. Usually, these things are done based on gut feelings, or you look at who closed the most issues, but here you can see much more detailed and valuable insights.

L: (Time bar ve time statistics gösterildi)

Katılımcı 2: Ya bu anormalliklerin içerisinde bulunan insanların sayısı, bu bulunma miktarı ve hangi insanlar olduğu olabilir.

**Participant 2:** The number of people involved in these anomalies, the frequency, and which people they are could be included.

U: Yani zaman içinde mesela hep Ali'yi işin odağında görüyorsak, nerede anomali var orada Ali var.

**U:** For example, if we always see Ali at the center of the issue, where there are anomalies, there is Ali.

Katılımcı 2: Bu bir red flag olabilir.

**Participant 2:** This could be a red flag.

Katılımcı 1: Değer üreteceğiniz şöyle bir alan daha olabilir. Özellikle build'lerin yanına veya deployment'ların yanına bunu bağlarsanız. Bu build'te temel olarak neler var, farklı olan neler commit'lendi, pull request'lerden hareketle yani user story giden. Şimdi o açıdan benzer çalışan şeyler var hatırlarsın. Onu görselleştirmiş oluyorsunuz. Belli bir değer çıkabilir oradan.

**Participant 1:** There could be another area where you can create value. Especially if you link this to builds or deployments. What is basically included in this build, what different things were committed, based on pull requests and user stories. Similar things are done in some aspects, if you remember. You are visualizing that. There can be some value derived from that.

E: Evet yani bu deploy'da ürünün şu kısımlarına dokunulmuş. Bu developer'lar involve olmuş. Bayağı güzel olur.

**E:** Yes, it would be great to see which parts of the product were touched in this deployment and which developers were involved.

Katılımcı 1: Şu pull request'ler, şu dosyalar gibi. O zaten geliştiriciler tarafından karşılığını bulur.

**Participant 1:** Like which pull requests and files. That would resonate with the developers.

E: Evet, evet güzel olur.

**E:** Yes, yes, it would be good.

Katılımcı 1: Hem de pipeline'ı direkt eklemiş olursunuz, gerçek zamanlı.

**Participant 1:** And you would directly add the pipeline, in real-time.

2:19:51 - 2:21:25

L: Tamamdır, şimdi isterseniz bu deneme bölümünde yaptığınız task'lar dolayısıyla verdiğiniz cevaplar üzerinden gidebiliriz

**L:** Okay, now if you want, we can go over the answers you gave during this trial section.

L: İlk sorunuzda veriler Any23.java dosyası için kimi expert olarak önerdiniz diye sormuştuk. 2 kişiyi tercih etmenizi istemiştik. Birinci Lewis çıkmış, GitHub'da ve bizim tool'umuzda ortak olarak öyle görünüyor. İkinci olarak Hans ve Peter eşit gelmiş. Bu konu hakkında, ikinciye Hans'ı seçen veya ikinci Peter'ı seçen bir katılımcı yorum yapmak ister mi acaba?

**L:** In the first question, we asked whom you recommended as an expert for the Any23.java file. We asked you to choose two people. Lewis came out as the first choice, appearing in both GitHub and our tool. For the second choice, Hans and Peter were tied. Would any participant who chose Hans as the second choice or Peter as the second choice like to comment on this?

Katılımcı 5: Ben en son commit edenlerden biri Peter olduğu için onu seçtim.

**Participant 5:** I chose Peter because he was one of the last committers.

L: Yani güncelliği değerlendirdiniz.

Katılımcı 2: Ben yanlış sonucu bulmuştum herhalde ama Michael'in bayağı commit'i vardı. Commit sayısı olarak bayağı fazlaydı, onu gördüm. O yüzden onu seçtim.

**Participant 2:** I think I found the wrong result, but Michael had quite a lot of commits. I saw the number of commits, so I chose him.

Katılımcı 1: Saydın mı bir şey durum ya?

**Participant 1:** Did you count or something?

Katılımcı 2: Evet, evet o şeyi açtım, oradan baktım, historisinden baktım.

**Participant 2:** Yes, yes, I opened it and looked at the history.

L: Daha eski olduğu için puanı daha düşük çıkıyordu.

**L:** Because it was older, his score was lower.

2:21:25 - 2:22:01

L: Diğer bir sorumuzda da aslında hangisinin daha güvenilir sonuçlar verdiğini sormuştuk. Burada da sanırım genel olarak insanlar recency value ile birlikte SAA'yı tercih etmiş. Ama genelde SAA tercih edilmiş. Bu konu hakkında yorum yapmak isteyen olur mu?

**L:** In another question, we asked which tool provided more reliable results. It seems that people generally preferred SAA, considering the recency value. Would anyone like to comment on this?

Katılımcı 1: Olamaz zaten biraz şey yani sonucu belli bir soru. Çünkü birisinde göz yanılması, insan çabası var, birisinde otomatik bir şey var. Evet, kendi kendini doğrulayan bir şey yani.

**Participant 1:** It's a somewhat obvious question because one involves human effort and potential errors, while the other is automatic. It's a self-confirming thing.

Katılımcı 4: Bir de GitHub'dan bakarken ben de şöyle bir sıkıntı oluyor, yani ben onu hissediyorum. Sanki her şeyi bakmadım mı? Bilerek bir tahmin yürüttüğümü biliyorum ama burada hani bana bir öneri geldiği için birazcık daha güvenilir geliyor. Öbüründe bir şey kaçırıyor muyum diye düşünüyorum.

**Participant 4:** Also, when looking at GitHub, I sometimes feel like I might have missed something. I know I'm guessing, but here, since I get a suggestion, it feels a bit more reliable. With GitHub, I worry if I missed something.

2:22:01 - 2:24:40

L: Diğer bir sorumuzda ise SAA hakkında inandırıcı bulmadıysanız nedenleri ne diye sormuştuk. Kompleksite çıkmış, bu konu hakkında yorum yapmak isteyen olur mu?

**L:** In another question, we asked why you found SAA unconvincing, if that was the case. Complexity came up. Would anyone like to comment on this?

Katılımcı 1: Bilmiyorum yani ben limitation takıldım ama kısa sürede öğreniriz diye düşündüğüm için taraf takılmadım.

**Participant 1:** I don't know, I was hung up on the limitations, but I thought we could learn it in a short time, so I didn't dwell on it too much.

E: Daha açar mısınız?

E: Could you elaborate?

Katılımcı 1: Onun dışında şey, yani ilk odaklıyız ya, bu sonuçta benim koymuş olduğum temel şeye göre kriterlere göre bir veri üzerinde çalışacak zaten. Yani bizim dünyalarında iş biçimimiz seçici, o da seçici bir veri kümesi ile çalışacağı için zaten başlangıcı gene ben seçmiş oluyorum. Aslında benim sınırlamalarım, benim doğru olmayanlar belirlememle beraber çalışmış oluyor. Araç kendisi her şeyi gezme ve keşfetme alanına sahip değil. Benim için bir şey olduğum sorular içerisinde kaldığı için beni devralmak zorunda. Birinci de nasıl kendinden emin değilsen burada sınırları ben belirlediğim için öyle bir şey var zaten.

**Participant 1:** Besides that, we're initially focused on a specific dataset, which works according to the criteria I set. Since our world is selective, and the tool works with a selective dataset, I'm essentially setting the initial parameters. My limitations, my potentially incorrect assumptions, are what it works with. The tool doesn't have the freedom to explore and discover everything on its own. Since it remains within the questions I have, it has to take over from me. Just like when you're not confident in the first result, here, because I set the boundaries, it inherently has that aspect.

E: Ama bütün araçlar böyle olur o zaman.

E: But then all tools would be like thatKatılımcı 1: Yani o sıfırdan başlayanlarla olmuyor biliyorsun yani her şeyi tutuyorsan.

**Participant 1:** It's not the same with tools that start from scratch and keep everything.

E: Tamam. Bias'la neyi kastettiniz? Niye bias?

E: What did you mean by bias? Why bias

Katılımcı 2: Onu ben söyledim, bir nevi ister istemez burada bir istatistik bilimi konuşuyor ve onunla ilgili şeyler olduğu için orada bu konuda bir bias'ın olması gerektiğini düşünüyorum. Çünkü daha önce çalıştığım projelerde bir gün içerisinde bütün projenin şeyini yazan kişi değişmişti. Çünkü farklı bir repoya taşındık ve oradaki bütün işi yapan kişi sadece bir kişi olarak görünüyordu. Böyle durumlar bu sefer böyle şeyler olabiliyor. Yani geçmişin silinmesi gibi durumlar olabilir. Bunlar olduğu için tamamen güvenmek doğru gelmedi.

**Participant 2:** I mentioned that. Inevitably, because we're talking about statistical science and related aspects, I think there's a bias. In previous projects I worked on, the person responsible for the entire project in one day changed because we moved to a different repo, and all the work appeared to be done by one person. Such situations can erase the past, making it unreliable.

E: Tamam okey, ama bu da yine yani herhangi bir tool için geçerli olan bir şey. Yani başka bir x aracı olsa o da aynı bias'a sahip olurdu.

**E:** Okay, but this would be true for any tool, right? Any other tool would have the same bias.

Katılımcı 2: Tabii ki, tabii ki.

**Participant 2:** Of course, of course.

2:24:40 - 2:25:12

**L:** Bir diğer sorumuzda ise GitHub ile bulma yönteminiz güvenilirmez ise neden diye sormuştuk. Burada da limited content işaretlenmiş, bu konu hakkında yorum yapmak isteyen olur mu?

**L:** In another question, we asked how much you think this visualization contributes to making the results understandable. According to the answers, we generally think it has a positive contribution. Can someone who marked 3 discuss this?

**E:** O zaman burada da limited content var değil mi?

**E:** So, limited content here as well, right?

Katılımcı 2: Evet, aynen, onu işaretlemiştim.

**Participant 2:** Yes, exactly, I marked that.

Katılımcı 1: Burası da benim için kompleks geliyor, bir yerden sonra dosya büyüklüğünü görüp kompleksleşiyor, insan sınırını aşıyor.

**Participant 1:** This also feels complex to me. At some point, the file size increases and it becomes complex, exceeding human limits.

2:25:12 - 2:26:11

**L:** Diğer bir sorumuzda da aslında genel olarak bu görselleştirmenin sonuçların anlamlandırılabilirliğine ne kadar katkı sağladığını sormuştuk. Cevaplara göre genel olarak pozitif katkı sağladığını düşünüyormuşuz. Bunu da mesela 3 diyen biri konuşabilir mi acaba?

**L:** In another question, we asked how much you think this visualization contributes to making the results understandable. According to the answers, we generally think it has a positive contribution. Can someone who marked 3 discuss this?

**E:** Ya da nasıl geliştirilebilir?

**E:** Or how it could be improved?

**L:** Herhangi bir konuşmak isteyen var mı?

**L:** Does anyone want to comment?

Katılımcı 1: Görselleştirme, her halükarda başlangıçta da söylediğiniz gibi, bizim algımızın daha güçlü olduğu bir alan. Sözlü yapıya göre ya da yazılı yapıya göre görsel yapı çok daha iyi. Yani galiba önce görsel sonra yazılı şeye sahibiz. Algıya sahibiz. Görselin her halde yüksek bir değeri var.

**Participant 1:** Visualization, as you initially said, is a stronger area for our perception. Compared to verbal or written structures, visual structure is much better. I think we first have visual then written perception. Visuals inherently have high value.

E: Tamam.

E: Okay

L: Tamam, diğer bir sorumuzda da aslında eklediğimiz bu görsel elementlerin, mesela node badgelerinin, anlayışa ne kadar katkı sağladığını sormuştuk. Bunda da genelde pozitif yanıt çıkmış. Bu konu hakkında konuşmak isteyen olur mu? Yani şu şekilde geliştirilebilir şeklinde de bir yorum alabiliriz.

**L:** In another question, we asked how much the added visual elements, like node badges, contribute to understanding. The responses were generally positive. Would anyone like to comment on this? We can also take suggestions on how to improve it.

Katılımcı 1: Görselliğin, yani bütün güzelliğine karşı, aslında daha önce farklı yerlerde başladı. Birkaç defa geldi belki de sizin de gelmiştir hatırlarsınız. Yani normalde böyle görsel olarak her şey özetleme büyük geliyor ama bir anda çalıştığımız veri kümesinin ne kadar büyük olduğunu fark edip yeniden daralma ihtiyacı var. İşte trace'in gitti mesela, bütün projeleri ilk olarak karşınıza koyup hiçbir şey anlayamazsınız. Mümkün değil, o kadar büyük bir yapı ortaya çıkıyor ki. Yani işte burada basit 7.000 tane not vardı, 23 bin ilişki vardı. Bir projede bir sene sonra o sayıya ulaşırsınız zaten çok rahat bir şekilde. Üçüncü senesinde falan ya da ikinci senesinde sayılar çok da büyük hale geliyor. Onu gösterecek ilgi odaklı bir yapıya dönüştürmeniz lazım.

**Participant 1:** Despite the beauty of visuals, it actually started in different places before. Maybe you've encountered it a few times as well. Normally, summarizing everything visually seems great, but suddenly realizing the vast size of the dataset we're working with makes us need to narrow it down again. For instance, when you first lay out all the projects in the trace, you can't understand anything. It's impossible because such a large structure emerges. Here, for instance, there were 7,000 nodes and 23,000 relationships. In a project, you easily reach that number within a year. By the third or even the second year, the numbers become very large. You need to transform it into a structure focused on interest.

Katılımcı 1: Aslında sorun orada, ilgi odaklı görselleştirme. Yani hem insan algısının baş edebileceği hem ilgi çekebilecek gösterim. Bu kadar güzel diyorum çünkü her şeyi koyarsanız zaten hiçbir şey anlaşılmayan bir yapı ortaya çıkıyor.

**Participant 1:** Actually, the issue is focused visualization, something that both human perception can handle and can be engaging. I'm saying it's beautiful because if you put everything together, it becomes an incomprehensible structure.

L: Aslında yani bizim de bu complexity management ile ilgili koyduğumuz şeyler, ana noktada amacımız bahsettiğiniz bu caselerin önüne geçebilmek.

**L:** Actually, our main goal with the complexity management features we added is to address these cases you're mentioning.

Katılımcı 1: Dashboard orada iyi olabilir, yani yönelim sağlayabilir. Yani ilgi odaklı bir görselleştirme ile beraber benim algımın, baş edebilmesini istiyorum. Bir de bu zooming var ya, yakınlaşma uzaklaşma, yakınlaştıkça şeyler artar, expand eder. O şeyler, grupların, klasların ya da yapıların uzaklaştıklarında toplanması algıyı daha rahat yönetir.

**Participant 1:** A dashboard could be useful there, providing direction. I want my perception to be able to handle it with interest-focused visualization. And there's the zooming feature, where things expand as you zoom in. When you zoom out, groups, classes, or structures collapse, making it easier to manage perception.

E: Tamamdır.

**E:** Understood.

Katılımcı 1: Onu optimize etmek gerekirdi diye düşünüyorum.

**Participant 1:** I think it needs to be optimized.

2:28:02 - 2:32:20

L: Diğer bir sorumuzda aslında SAA ile expert bulmanın sizin normalde kullandığınız metoda kıyasla daha kolay olup olmadığını sormuştuk. Genellikle kullanıcılar daha kolay olduğunu düşünmüş.

**L:** In another question, we asked if finding experts with SAA was easier compared to the method you normally use. Generally, users thought it was easier.

E: Şimdi bunu geçelim.

**E:** Let's move on.

**L:** Geçiyoruz. Bu da zaten sanırım genelde herkes üç kişiyi bulmuş, bir kişi büyük bulmuş. Bunlar hakkında konuşmak isteyen var mı yoksa direkt geçebilirim. Sonuçlarımızın nasıl bir deneyimdi sizce? Kolay mıydı değil miydi?

**L:** Moving on. It seems that most people found three experts, one person found more. Does anyone want to comment on this or shall we move on? What was your experience with our results? Was it easy or not?

**Katılımcı 4:** Ya JQL bence genel olarak çok zor, hiç user friendly değil. Onun için bunun bir alternatif olması şahsen benim hoşuma gitti.

**Participant 4:** Well, JQL is generally very difficult, not user-friendly at all. So personally, I liked having this as an alternative.

**Katılımcı 1:** Zaten Jira onu çok eskiden beri kullanıyor ve vazgeçmiyor. O şey, ta öncelerden, Bugzilla'dan falan gelen bir özellik. Ta eskilerden SQL yazmak zaten sadece developer'ların kullandığı bir araç olduğu için o dönemde ama sonra işte görsel anlatım ya da SQL'in böyle daha yapısal editörleri falan öne çıkmıştı ama Jira oradan developer'lardan dolayı vazgeçmedi. Sevenler de çok, mesela hocam da seviyordu, çok seven de var o haline yani onu mesela kapatsan isyan çıkar.

**Participant 1:** Jira has been using that for a long time and hasn't let go of it. It's a feature from long ago, from Bugzilla, because SQL was only used by developers back then. Later, visual representation or more structured SQL editors became prominent, but Jira didn't move away from it because of developers. Many people love it, for instance, my professor loved it. If you shut it down, there would be an uproar.

**Katılımcı 1:** Fakat dışarıdan başladığınız zaman da zor, anlamlı değil. Yani biliyorsun ki niye şimdi böyle bir şeyle işkence çekiyoruz? Seçmek için çok daha kolay seçenek de olabilir falan diyorsunuz. İlişkisine göre.

**Participant 1:** But starting from scratch is hard and meaningless. You wonder why you're suffering with this. There could be much easier options for selection based on relationships.

**L:** Tamamdır, geçiyoruz.

**L:** Okay, moving on.

**L:** Mesela bazı caselerde JQL kullanılarak bulunamayan anomallerimiz var. Mesela son anomaly aslında öyleydi. Yani direkt resolver bölümüne direkt eden bir field olmadığı için JQL ile zaten bulamıyorduk, bulamadık sanırım.



**L:** In some cases, we have anomalies that can't be found using JQL. For instance, the last anomaly was like that. Since there wasn't a field directly pointing to the resolver section, we couldn't find it with JQL.

L: Şimdi devam edelim.

**L:** Let's continue.

L: Diğer bir sorumuzda aslında, yüksek sayıda anomaly'ye sahip bir issue bulup daha sonrasında Jira'dan gidip bu issue'nun paneline gidip o panelden inceleme yapıp kaç anomaly görebiliyorsunuz, bunu yazmanızı istemiştik.

**L:** In another question, we asked you to find an issue with a high number of anomalies, go to that issue's panel in Jira, and write down how many anomalies you could see.

E: Bu da bir şey durumu olduğunu, cevabı verince daha kolay anlaşılmış olabilir. Öyle bir durum olmuş oldu da.

**E:** This might have been easier to understand after answering. That's how it seemed.

L: Zaten sanırım herkes çoğunu bulabilmiş. Onun dışında bir kişi sanırım hiçbirini bulamamış.

**L:** It seems most people found most anomalies. Except one person who couldn't find any.

Katılımcı 1: Benimdir o ya, ben duruyor ya normalde bakınca. Ya bununla ilgili mi yok zaten?

**Participant 1:** That must be me. Normally, it seems like there isn't anything related to this.

L: Anladım.

**L:** I understand.

L: Son sorumuzda da aslında Jira ve SAA'yı karşılaştırdığımızda anomaly detection açısından hangisinde daha rahattı diye sormuştuk ama zaten konuştuk. Üstüne daha fazla konuşmak isteyen olursa.

**L:** In our last question, we asked which was easier for anomaly detection, Jira or SAA, but we've already discussed this. If anyone wants to comment further.

Katılımcı 2: Ya ben tek tek hepsine bakmaya çalıştım, geride tutturamadım yani oradaki şeyleri.

**Participant 2:** I tried to look at each one, but I couldn't keep up with the items there.

L: Tamamdır. Yine en son sorumuzda daha kolay mıydı, zor muydu? Normalde kullandığınız bir practice ile kıyasladığınızda diye sormuştuk. Burada yaklaşık 3 kişi çok daha kolay olduğunu, 2 kişi de böyle bir metodu kullanmadığını zaten belirtmiş.

L: Okay. In our final question, we asked if it was easier or harder compared to the method you normally use. About three people said it was much easier, and two said they don't use such a method.

Katılımcı 1: O mavi hangi mavi, %40 olan?

**Participant 1:** Which blue is that blue, the one that's 40%?

L: Hocam, o daha önce böyle bir tool kullanmadım.

L: I haven't used such a tool before.

Katılımcı 4: Ha, diğer türlü çok ilginç olurdu.

**Participant 4:** Oh, otherwise it would be very interesting.

U: Burada size bayağı göndermemiz gerekebilirdi.

U: We might have needed to send you quite a bit here.

L: O zaman başka yorumu olan yoksa devam edeyim? Son bir sorumuz kaldı.

L: If no one else has a comment, shall I continue? We have one last question left.

2:32:20 - 2:36:00

L: Genel olarak SAA hakkındaki deneyiminizi sormak istemiştik. Kafa karıştırıcı noktaları neydi sizce veya siz bu aracı kullanıyor olsaydınız en fazla hangi use case bazında kullanıyor olurdunuz? Hani inspection kullanırdınız yoksa yeni bir analiz mi entegre ederdiniz, yoksa hâlihazırda olan analizlerden birini mi kullanırdınız? Bu şekilde bir soru sormuştuk.

L: We wanted to ask about your overall experience with SAA. What were the confusing points, and if you were using this tool, for which use case would you use it the most? Would you use it for inspection, integrate a new analysis, or use one of the existing analyses? This was the question.

Katılımcı 3: Skor kısmını kesin kullanırdım (expert, reviewer part). Orası benim en çok hoşuma giden yerlerden birisi. Hatta daha da gelişebilir bence o kısımlar. Bu tabii anomaly detection kısımlarına da bakardık, onu ara ara kullanmak isterdik ama bunu bize raporlasa daha iyi olurdu. Çünkü ben sürekli açıp bakamam. Oraya mesai harcamadan, bana raporlasa "Ha öyle mi?" deyip bir detayına bakmak isterdim. O kısım benim hoşuma gitti. Bir de gerçekten hani çok, görsel açıdan yani ne kadar çok nesne olsa da içine girdikçe yeni yeni kapılar açıyor size. Bir ağaç oluşuyor gibi. Belli bir noktadan sonra ipin ucunu kaçırabilirdik. Yani bazı işlerimiz

gerçekten çok detaylı oluyor ama belli noktada da gerçekten işimizi çok rahatlatır. "Ha öyle mi? Sonra ne olmuş? Kim review etmiş? Ne sıkıntı yaşamış?" falan görmek açısından görsel açıdan da gerçekten faydası olurdu, onu da kullanırdık büyük ihtimalle.

**Participant 3:** I would definitely use the scoring part (expert, reviewer part). That's one of my favorite areas. I think it could be further developed. We would also look at the anomaly detection parts from time to time, but it would be better if it reported to us. Because I can't keep checking it all the time. If it could report to me without spending time on it, I would look into the details. I liked that part. And visually, even though there are many objects, it opens new doors as you dive in. It's like a tree forming. At a certain point, we might lose track of it. Some of our tasks are very detailed, but at some point, it would really make our job easier. "Oh, really? What happened next? Who reviewed it? What issues were there?" Visually, it would be really useful, and we would probably use it.

Katılımcı 3: Genel olarak böyle ama bence en güçlü olduğumuz. Burada belki hiç konuşulmayan bir nokta ama, yani bütün bu veriyi alıp bir database oluşturmanız ve onu görselleştirebilmeniz gerçekten güzel bir özellik. Bu database'i belki farklı noktalarda da evrilleştirebilirsiniz, geliştirebilirsiniz. Mesela bir pandas gibi bir veri işleme tool'u içerisinde oradaki database'i değerlendirip farklı dashboard'lar oluşturabilme, onları gösterme, belki bunu kullanıcının yapmasına izin verme.

**Participant 3:** Overall, that's it, but I think our strongest point, which might not have been discussed here, is being able to take all this data, create a database, and visualize it. It's a really nice feature. This database could be evolved and developed at different points. For example, using a data processing tool like Pandas to evaluate the database, create different dashboards, show them, and maybe allow the user to do this.

Katılımcı 3: Hani burada SQL yazıyoruz da orada da kendimize ait bir dil olsa da, mesela ben onun üzerinden bir şeyler yapabilsem dedim. Fikir sadece tabii, o çok daha detaylı kapsamlı konular, onun detayını çok bilemem ama o tarz yapılar belki güzel olabilirdi. Tabii.

**Participant 3:** We write SQL here, but having our own language there, for instance, if I could do things with it, would be nice. It's just an idea, of course, very detailed and comprehensive topics that I can't fully detail, but such structures could be good. Of course.

L: Çok teşekkür ederiz.

L: Thank you very much.

Katılımcı 2: Benim için ve dikkatimi çeken kısım bu anomaly detection bölümüydü. Daha çok, yani muhtemelen commit ile ilgili olan kısımlar. Çok fazla işimize yarar mı, emin değilim. Çünkü bir projedeki şeyin o commit ile ölçülmesine genel olarak karşı olduğum için falan gibi bir durum var. Anomaly kısmı bence çok değerli bir kısım. Orayı kesinlikle kullanırdım diye düşünüyorum. Bir de o kısımların aslında arka planının nasıl çalıştığını, yani oralara tam anlamadım, anlamak isterdim. Daha doğrusu ve en başta bahsetmişsiniz. Buraya bir şey eklemek istediğimizde şunları kullanıyoruz, bunu şöyle böyle yaparız diye bahsetmişsiniz. Belki oralar biraz daha açık

olabilirdi. Ekleme için yani yeni şeyler entegre etmenin daha kolay olduğu bir dünya daha iyi olabilir diye düşünüyorum. Belki kolaydır bu arada, ben anlamamış olabilirim.

**Participant 2:** For me, the part that caught my attention was the anomaly detection section. More specifically, probably the parts related to commits. I'm not sure how much it would benefit us because I'm generally opposed to measuring something in a project based on commits. I think the anomaly section is very valuable. I would definitely use that. Also, I didn't fully understand how the backend of those parts works. You mentioned at the beginning about adding things and how you would do it. Maybe those areas could be a bit clearer. I think a world where it's easier to integrate new things would be better. Maybe it's already easy, and I just didn't understand it.

U: Biz zaten bir tane developer guide yazdık. Yani birileri gelip de bu aracı alır ve yeni bir analitik metodu eklemek isterse hangi steplerden geçmesi gerektiğini döküman ettik. Dolayısıyla o zor bir şey değil.

U: We actually wrote a developer guide. So if someone comes and wants to add a new analytical method to this tool, we've documented the steps they need to take. Therefore, it's not difficult.

L: Bugün burada çok ayrıntılı bir anlatım yapamadım o bölümle ilgili.

L: I didn't give a detailed explanation of that section today.

Katılımcı 2: Tabii, tabii doğru. Evet, doğru.

**Participant 2:** Of course, that's true.

Katılımcı 1: Ben de şöyle düşünüyorum, normal olayları değerlendirirken perspektif bazlı düşünmek fayda getirebilir. Burada bir değer var. Değerin kimin için ne ifade ettiğini, gerçekten kimin hayata ne katacağını biraz şekillendirmek lazım. O zaman değer olup olmadığını gerçekten anlayacağız. Yani bulgu değer midir, yoksa bulgu olarak mı kalır. Bu da perspektif bazlı bakarsak, hakikaten şey, endüstriyel rollere göre, belki farklı farklı alanlarda ve farklı farklı kullanıcıların pozisyonuna göre iyi olabilir. User story'den başlayıp build ve deploy'a kadar gitmek lazım. Ürün yöneticisi için, geliştirme takımı için, geliştiriciler için, test tarafı için ya da DevOps tarafı için, herkes için ayrı içgörüler, insight'lar veren şeyler ortaya çıkarmak. Çünkü herkes için aynı şey, bir şey herkeste aynı değer oluşturulmaz ama farklı farklı bakışlara göre sunabileceğiniz değer kümeleri var. Bakışlara göre sunarsanız fayda yükselir.

**Participant 1:** I also think that when evaluating normal events, thinking from a perspective-based approach can be beneficial. There's value here. We need to shape what the value means to whom and who will actually benefit from it. That's when we'll truly understand if it's valuable. Whether a finding is valuable or just remains a finding. If we look at it from a perspective-based view, it could be good for different industrial roles, in different areas, and according to the positions of different users. We need to go from the user story to the build and

deploy stages. Creating insights for the product manager, the development team, developers, the test side, or the DevOps side. Because the same thing won't have the same value for everyone, but there are value sets you can present from different perspectives. If you present it from different perspectives, the benefit increases.

E: Teşekkür ederiz.

E: Thank you.

L: Bu kadar, zaten başka bir sorumuz yoktu. Çok teşekkür ederiz. Sende bir sorun yok, anladın.

L: That's all, we didn't have any other questions. Thank you very much. You understood, no problem.

Katılımcı 1: Polislik yerine doğal olarak sağlık tarafından yenilik daha iyi. Aslında bu projenin ya da geliştirmenin sağlığı gibi, ya da ürün sağlığı gibi sağlık sözcükleri daha güzel oluyor. Yoksa finding ve policy tarafı biraz daha ürkütücü, korkutuyor. Herkes kaçır oradan. Commit sayısı ya da değişiklik sayısı ya da adam saat gibi değerler genelde insanları rencide ediyor.

**Participant 1:** Instead of policing, it's naturally better to approach it from a health perspective. Terms like the health of the project or the health of the development are better. Otherwise, the terms finding and policy can be a bit intimidating and scary. Everyone would avoid that. Values like the number of commits, the number of changes, or man-hours usually embarrass people.

E: Bir de bir tane survey daha göndereceğiz, onu da doldurursanız seviniriz. Onu sonra ile gönderecek. Çok teşekkür ederiz. Çok katkıda bulundunuz. Çok sağ olun, çok güzel feedback'ler aldık, kullanacağız. Çok sağ olun.

E: We'll also send another survey, and we would appreciate it if you could fill it out. We will send it later. Thank you very much. You contributed a lot. We received great feedback that we will use. Thank you very much.