| Quality Attribute | Quality Attribute Scenario | Tatic | Justification |
| --- | --- | --- | --- |
| Usability | Team collaborators make or receive an audio call with continued use and no dropouts until the call is ended by a user. | Cancel, Maintain a system model | The cancel action can map to the user ending the call.  By maintaining a system model the system will be able to detect if continued use or drop outs are occurring and present that information to the user. Additionally when those dropouts are detected the system can take action to prevent a significant drop out or further dropouts. |
| Usability | Team collaborators start or join video conferencing with continued use and no interruption of bandwidth or disconnection until the conference is ended by a user. | Cancel, Maintain a system model | The cancel action can map to the user ending the call.  By maintaining a system model the system will be able to detect if continued use or drop outs are occurring and present that information to the user. Additionally when those dropouts are detected the system can take action to prevent a significant drop out or further dropouts. |
| Usability | Team collaborator sketches or draws in the software application interface during normal operation. The user interface should respond, accept, and display user input within 300 milliseconds. | Maintain User Model | Maintaining a user model will help a lot to achieve this quality attribute scenario because it involves a lot of user interaction. By maintaining a user model the system will be able to accurately determine a real user touch from a ghost touch, as well as, reshape user drawn lines and letters for smoothness. |
| Performance | A collaborator uploads a file to the collaboration and the file is of the correct size. The system keeps the file unmodified and not corrupted. All collaborators can see the file and access it. | Introducing concurrency | Concurrency can be introduced because uploads are not dependent on other uploads and accesses are not dependent on other accesses and can therefore be processed concurrently. This will increase performance time by dividing the current run time by the number of available concurrent threads. |
| Performance | A collaborator uses email to collaborate and add information to a work group. An email is sent to a work group and is received by users without data corruption or loss. | Introduce concurrency | Concurrency can be introduced because messages sent to distinct receivers and from distance sender are not dependent on other messages and can therefore be processed concurrently. This will increase performance time by dividing the current run time by the number of available concurrent threads. |
| Modifiability | Team collaborators are able to send output from the software application to an external display such as a smartphone at runtime. The software application modifies the external display with no downtime, no faults, side effects. | Encapsulate | The module creating the output can be encapsulated from the module displaying the output. Because the only knowledge that needs to be shared between the two modules is the image data. This will increase modifiability because any type of display module that is able to receive the data will be able to display the data. |
| Modifiability | Team collaborators are able to add a variety of input devices such as a wireless pen, tablet, or touchscreen at runtime to the software application. The software application adds the device and operates with no downtime, no faults, and no side effects. | Use an Intermediary | By using an intermediary the output device won't have to worry about sending the right output message to the right type of device. Instead the output device can put all of its messages onto an intermediary while other devices listen to that intermediary and can then act on the messages they are able to. This improves modularity because any device can now listen to that intermediary and act on the messages it has the ability to. |
| Usability | Team collaborators are able to launch the software application using the Safari, Chrome, or Firefox web browsers and have full functionality. It is used indefinitely with no error codes, no faults, and no crashes. | Maintain system model | A maintaid system model can be used to determine if there are any code errors, faults or crashes. When those errors are determined the system can take steps to handle those errors. |
| Usability | The user logs into the system for the first time. The system renders printed content in the users correct language based on their locale 95% of the time. | Maintain User Model | This software quality attribute basically describes a user model. Basically the system would model the user’s language based off of their location. And the app would be presented to the user in that language. |
| Security | A non-authorized employee attempts to access the software application during normal operations. The software application protects unauthorized access using SAML authentication and two step verification. The user login and IP address is blacklisted after 5 unsuccessful attempts. The software application identifies the source of the attack within a day. | Authorize Actors, Identifying actors | This scenario mentions SAML authentication and two step verification process therefore the authorizing Actors tactic will help to distinguish valid users from non-valid users, by only authorizing actors that have a valid SAML and two step verification. This scenario also mentions blocking people based on IP address, therefore Identifying actors tactic can be used as a means to check this IP address versus the existing deny list and determine if the IP address is still authorized to access. |
| Security | A non-authorized employee attempts to access the software application to obtain audio and video system data and services during normal operations. The software application protects these services from unauthorized access by encrypting audio and video data using TLS encryption and PKI certificates The software application identifies the source of the attack and restores services within a day. | Authorize Actors,  Limiting access, Encrypt data | The system can distinguish between the non-authorized employed mentioned and authorized employees by using a password and two factor authentication system. This scenario mentions an employee accessing data, therefore the data access can be protected using the limiting access tactic so that only the people who need to see the data will have access. And any other attempt to view the data will be reported. The scenario also mentions encrypting the data so the encrypt data tactic can be used to fulfill that requirement. |
| Security | A non-authorized employee attempts to access the software application to obtain user data, files and emails. The software application protects system data from unauthorized access by encrypting all media and data using AES-256 with keys managed by a key management system. The software application identifies the source of the attack and restores data and services within a day. | Authorize Actors,  Limiting access, Encrypt data | The system can distinguish between the non-authorized employed mentioned and authorized employees by using a password and two factor authentication system. This scenario mentions an employee accessing data, therefore the data access can be protected using the limiting access tactic so that only the people who need to see the data will have access. Any other attempt to view the data will be reported. The scenario also mentions encrypting the data so the encrypt data tactic can be used to fulfil that requirement. |
| Security | A user launches the software application and accesses the software application and its data and services during normal operations. The software application tracks activities within it by recording login, recording attempts to access or modify data and services, and notifying system administrators of any suspicious activity. The source of tampering is identified within an hour. | Identify Actors, Detect Intrusion, Maintain audit trail, | This scenario mentions users with suspicious activity being identified, therefore the identifying actors tactic should be used so the malicious user can be distinguished from normal users.  This scenario also mentions tracking activities therefore the Maintain audit trail tactic will be helpful to store all of that activity data for each user. Finally, the scenario mentions notifying system administrators of suspicious activities, therefore the detect intrusion tactic should be used to determine if user activity is malicious or has a potential for damage. |
| Availability | The web browser user interface notifies team collaborators if the software application is overloaded and the bandwidth rate is below 5/5 MBPS. Ongoing video conferences are switched to audio only mode the software application continuously tries to re-establish connection. Video is disabled until the bandwidth exceeds 5/5 MBPS. | Monitor,  Removal of service, Rety | This scenario mentions a threshold of 5/5 MBPS affecting the functionally of the system therefore the monitoring tactic can be used to determine if that threshold is crossed. This scenario also mentions switching from video to audio mode in the event of low MBPS therefore the removal of service can be used to remove the video data and just use audio to prevent further errors. Finally the scenario meintions trying to re-establish connection, therefore the retry tactic can be used to retry connecting until the connection succeeds. |
| Availability | A heartbeat monitor determines that the software application is unresponsive at startup or during normal operations. A system administrator can identity and troubleshoot the error within 30 minutes and notify the user when application functionality is restored to 99.99%. | Heartbeat,  Self-test | This scenario mentions a heartbeat monitor therefore the heartbeat tactic should be used to fulfill this requirement. Additionally, the scenario mentions the ability for a system administrator to be able to identify and troubleshoot the error and notify the user when the functionality is restored to 99.99%. Therefore the self-test tactic should be used so the system can test itself and identify areas where the error are coming from and determine the functionality level. |