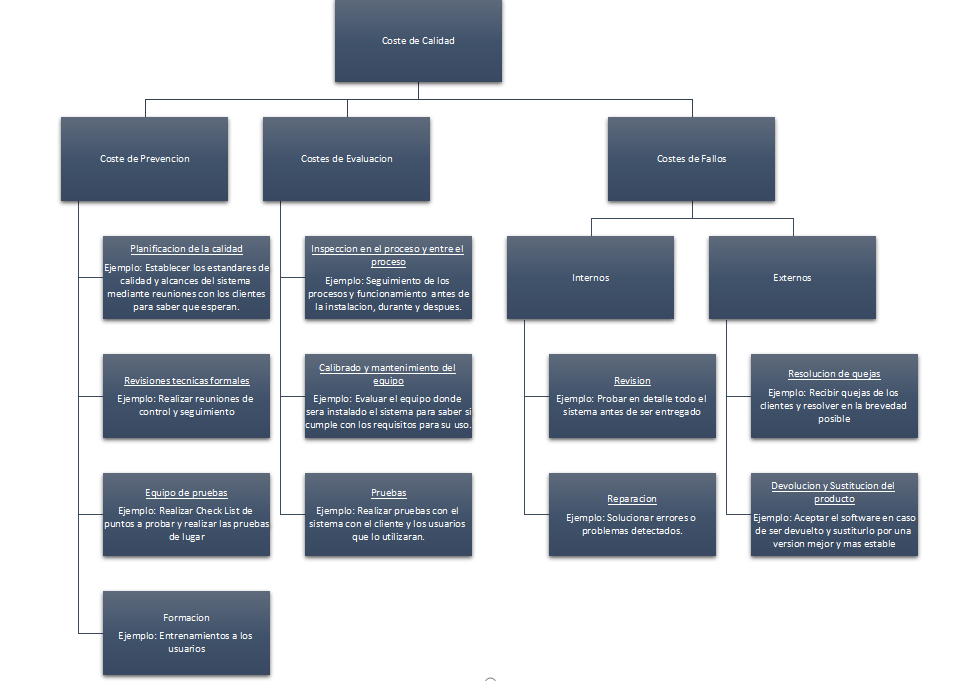
**TAREA 2**.- ASEGURAMIENTO DE CALIDAD (SQA)

1. **CONCEPTOS PRÁCTICOS**

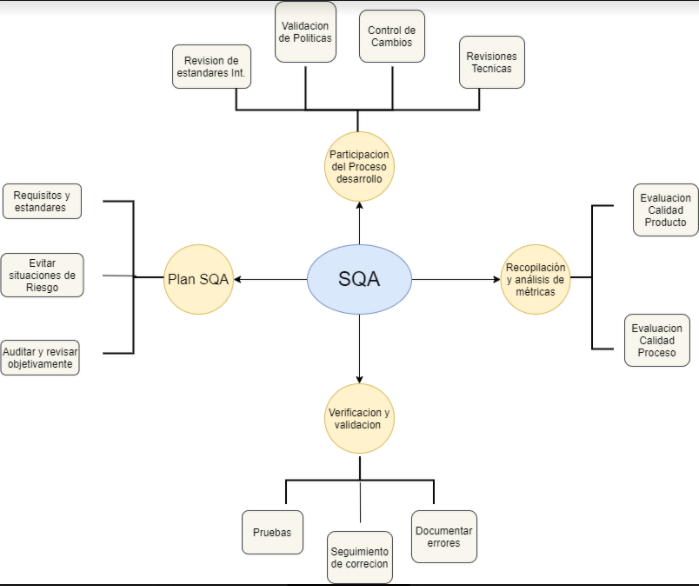
Fuente: Ing. De Software – Un Enfoque Práctico (Pressman), 5ta. Ed. Cap. 8 – Garantía de Calidad del Software

**Documento ‘Control y Gestión del Aseguramiento de la Calidad del Software’, disponible en el grupo web de la materia**

1. Elabore un esquema gráfico que contenga los diferentes costes de calidad, sus componentes y un ejemplo de una actividad que se incluya en estos últimos para cada uno.



1. Elabore un esquema gráfico que resuma las actividades de SQA y sus objetivos.



%3CmxGraphModel%3E%3Croot%3E%3CmxCell%20id%3D%220%22%2F%3E%3CmxCell%20id%3D%221%22%20parent%3D%220%22%2F%3E%3CmxCell%20id%3D%222%22%20value%3D%22%22%20style%3D%22endArrow%3Dclassic%3Bhtml%3D1%3BfontSize%3D12%3BfontColor%3D%23F0FFF8%3BexitX%3D0.656%3BexitY%3D-0.041%3BexitDx%3D0%3BexitDy%3D0%3BexitPerimeter%3D0%3B%22%20edge%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20width%3D%2250%22%20height%3D%2250%22%20relative%3D%221%22%20as%3D%22geometry%22%3E%3CmxPoint%20x%3D%22443.72%22%20y%3D%22506.72%22%20as%3D%22sourcePoint%22%2F%3E%3CmxPoint%20x%3D%22480%22%20y%3D%22440%22%20as%3D%22targetPoint%22%2F%3E%3C%2FmxGeometry%3E%3C%2FmxCell%3E%3C%2Froot%3E%3C%2FmxGraphModel%3E

1. Conteste las preguntas 8.1 – 8.11 y 8.14 – 8.15

## ***8.1. Al principio del capítulo se señaló que «el control de variación está en el centro del control de calidad». Como todos los programas que se crean son diferentes unos de otros, ¿cuáles son las variaciones que se buscan?*** Las que se buscan van desde los recursos, planificación y tiempo de desarrollo incluyendo así la complejidad y cohesión para el desarrollo del software.

## ***Y cómo se controlan?*** Se debe implementar los procesos de aseguramiento de calidad del software entre los cuales somos dependientes de los estándares pre-establecidos por la empresa, basandose en el tamaño de esta para así validar que cuenta con todos los requisitos, funciones y características previamente propuestas.

***8.2. ¿Es posible evaluar la calidad del software si el cliente. no se pone de acuerdo sobre lo que se supone que ha de hacer?***

No, no es posible. Esto ya que el cliente es que indica las funciones del software y no se puede evaluar si aun no esta de acuerdo.

***8.3. La calidad y la fiabilidad son conceptos relacionados, pero son fundamentalmente diferentes en varias formas. Discutirlas.***

Ambos estan relacionados, porque evalúan el proyecto y determinan sus parámetros. La diferencia entre ellos es que la calidad se evalua en base a que los requisitos cumplan los estandares de la empresa, mientras que la fiabilidad se trata de un buen funcionamiento.

***8.4. ¿Puede un programa ser correcto y aún así no ser fiable? Explique por qué.***

No, debido a que la fiabilidad va relacionado con el buen funcionamiento en un ambiente determinado e inmediatamente el mismo falle, no es fiable.

***8.5. ¿Puede un programa ser correcto y aun así no exhibir una buena calidad? Explique por qué*.**

No, debido a que un programa de calidad debe ser funcional cumpliento todos los parámetros y requisitos previamente establecidos, y si el mismo no los cumple implica que no posee calidad, por lo cual no es funcional para la empresa.

***8.6. ¿Por qué a menudo existen fricciones entre un grupo de ingeniería del software y un grupo independiente de garantía de calidad del software? ¿Es esto provechoso?***

En la ingeniería de Software se encargan de la elaboración del producto mientras que el grupo independiente de garantía de calidad se encargan de la evaluación del correcto funcionamiento, lo que implica que entre ambos habrán fricciones.

Sí, esto es provecho, debido a que las fricciones entre ambos llevan a debates y diferentes posturas, lo cual consigo trae ideas de soluciones e implementación de nuevos alcances y arreglos de errores.

***8.7. Si se le da la responsabilidad de mejorar la calidad del software en su organización. ¿Qué es lo primero que haría? ¿Qué sería lo siguiente?***

Lo primero que se tomaría en cuenta sería la aplicación de garantía de la calidad del software y luego de esto la verificación de la fiabilidad del mismo.

***8.8. Además de los errores, ¿hay otras características claras del software que impliquen calidad? ¿Cuáles son y cómo se pueden medir directamente?***

La fiabilidad, se mide en base a los datos estadísticos e históricos.

La seguridad, que se mide en base a la identificación de los riesgos y la determinación de la gravedad y probabilidad de ocurrencia.

Tiempo de respuesta, en base al funcionamiento y velocidad de realización de procesos.

***8.9. Una revisión técnica formal sólo es efectiva si todo el mundo se la prepara por adelantado. ¿Cómo descubriría que uno de los participantes no se la ha preparado? ¿Qué haría si fuera el jefe de revisión?***

Realización de preguntas de sondeo aleatorias en donde tendrían que estar en acuerdo o desacuerdo con casos que hipotéticamente sucedieron y la soluciones a los mismos, quienes estén de acuerdo indican que no están preparados, esto implica que deberá retirarse.

***8.10. Algunas personas piensan que una RTF debe evaluar el estilo de programación al igual que la corrección. ¿Es una buena idea? ¿Por qué?***

No, debido a que cada programador posee un estilo diferente, pero si deberían de tomar en cuenta el emplear un código entendible y plasmar el conocimiento del código.

## ***8.11. Revise la Tabla 8.1 y seleccione las cuatro causas vitales de errores serios y moderados. Sugiera acciones correctoras basándose en la información presentada en otros capítulos.***

* IEE (Especificación incompleta o errónea)
  + Sugerencia:
    - Aplicar métodos de recolección de datos como reuniones TFEA.
    - Programar preguntas específicas.
* MCC (Mala interpretación de la comunicación con el cliente)
  + Sugerencia:
    - Establecer minutas al finalizar las reuniones, en donde ambas partes han de estar acordes con lo planteado.
* ERD (Errores en la representación de los datos)
  + Sugerencia
    - Aplicar modelado de datos.
    - Depuración de los datos
    - Aplicar herramientas de desarrollo CASE
* PIE (Prueba incompleta o errónea)
  + Sugerencia
    - Verificación de funcionamiento de procesos.
    - Documentación de errores.

Depuración y nuevas implementaciones de errores.

**8.14. El concepto de TMEF del software sigue abierto a debate. ¿Puede pensar en algunas razones para ello?**

El TMEF, conocido como tiempo medio entre fallos, esto puede llegar a ocurrir cuando los usuarios no determinan los errores a tiempo y cuando se dan cuenta del mismo, ya hay un TMEF muy alto; estos no poseen un gran impacto en la fiabilidad del software.

**8.15. Considere dos sistemas de seguridad crítica que estén controlados por una computadora. Liste al menos tres peligros para cada uno de ellos que se puedan relacionar directamente con los fallos del software.**

* **Sistema de control de tránsito.** 
  + Se proyecta en verde simultáneamente para varias calles que coinciden en dirección.
  + Pierde el funcionamiento.
  + No poseen fuente de energía alterna.
* **Sistema de atención a emergencia.**
  + El sistema no redirecciona al agente disponible más cercano.
  + La llamada no toma en cuenta la geolocalización del usuario que llame.
  + No provee tiempo de respuesta a tiempo.

1. Elabore un esquema resumido de un plan SQA (Para el proceso final de Ing. De Software I), con un ejemplo para cada uno de sus componentes (documentos, requisitos, procedimientos, etc.). Como fuente adicional, puede utilizar el documento CONTROL Y GESTION DEL ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE.

En al adjunto llamado **Plan\_SQA\_TDT**

1. Organice y simule en el aula una RTF donde aplique cada una de las recomendaciones para la revisión, aplicada a este mismo proyecto. Puede elegir otros alumnos como colaboradores. Opcionalmente, puede entregar un video que contenga la actividad.