|  |  |  |
| --- | --- | --- |
| **Título componente** | **Acordeón** | |
| **Título** |  | |
| **Texto descriptivo** | Mera (2016), lista los siguientes principios: | |
| **Título** | **Texto** | **Imagen de referencia (opcional)** |
| **Las pruebas muestran la presencia de defectos** | Las pruebas son herramientas que facilitan detectar defectos, no obstante, no garantizan que no hayan defectos ocultos y no es una evidencia que el *software* esté óptimo. |  |
| **Pruebas tempranas** | Encontrar los errores en etapas tempranas, permite un importante ahorro de recursos. |  |
| **No es posible realizar pruebas exhaustivas** | Hay que realizar un análisis de riesgos, para identificar prioridades y tener una óptima toma de decisiones, centralizando esfuerzos y utilizando el talento humano y recursos de manera correcta. |  |
| **Las pruebas dependen del contexto** | Estas dependen del contexto en el cual se ejecutan, por lo cual, se debe dar más énfasis en las que sean para un sistema crítico, como un *software* financiero, para el cual se requiere realizar más pruebas, en comparación con otras aplicaciones con nivel críticos bajos. |  |
| **Agrupación de defectos** | Las pruebas deben agruparse, la mayoría de fallos operativos se enfocan en un número reducido de módulos. |  |
| **La paradoja de los pesticidas** | No hay que repetir la misma prueba varias veces, por lo cual es importante que los casos de prueba se revisen constantemente y escribir nuevos, con el fin de encontrar más defectos. |  |
| **Ausencia de error** | De nada sirve corregir en el sistema si este no cumple con los requerimientos o necesidades del usuario. |  |