



---

# PRINCIPIOS DE PREVENCIÓN DE DEFECTOS

---

By: Jared Isaías Monje Flores



3 DE SEPTIEMBRE DE 2023

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS  
UdeG

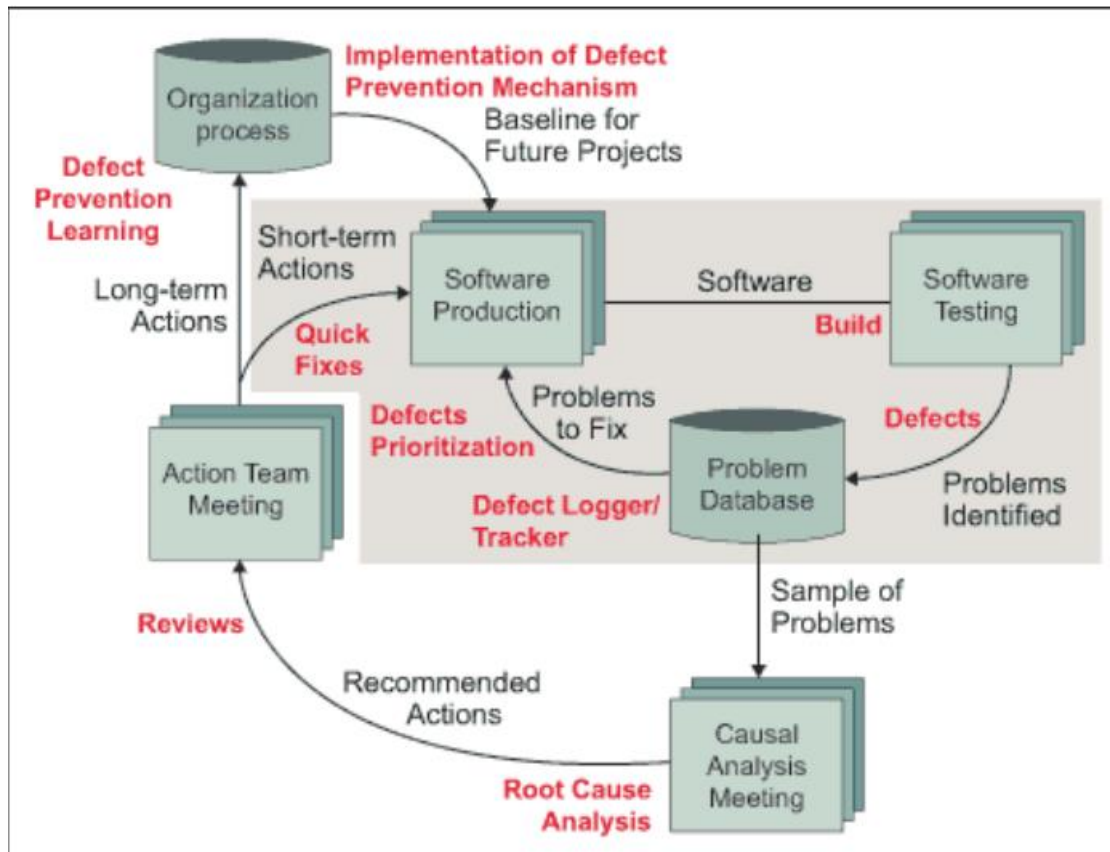
## **Principios de prevención de defectos**

La prevención de defectos es una tarea de grupo. Cuando una organización adopta una estrategia de prevención de defectos, sigue analizando y actuando sobre los datos de defectos directamente con los miembros de su grupo de desarrollo. Este indica con exactitud el coste de los errores del grupo y desafía al grupo a evitar los defectos, asumiendo la responsabilidad de la calidad del producto. Esta práctica da como resultado un diseño económico de alta calidad y motiva la recopilación de calidad. Crea un sistema competente en el que se utiliza continuamente para optimizar los procesos de diseño y comprobación.

La forma más eficaz de gestionar los defectos es prevenir su introducción inicial. En el PSP hay tres formas diferentes, pero mutuamente de prevenir los defectos. La primera consiste en que los ingenieros registren los datos de cada defecto que y solucionan. A continuación, revisan estos datos para determinar la causa de los defectos y proceso para eliminar estas causas. Al medir sus defectos, los ingenieros son más conscientes de sus errores, son más sensibles a las consecuencias y disponen de los datos necesarios para evitar cometer los mismos errores en el futuro. El rápido descenso inicial del total de defectos durante los primeros cursos de PSP indica la eficacia de este método de prevención.

El segundo método de prevención consiste en utilizar un método de diseño y una notación eficaces para diseños completos. Para registrar completamente un diseño los ingenieros deben comprenderlo a fondo. Esto sólo produce mejores diseños errores de diseño.

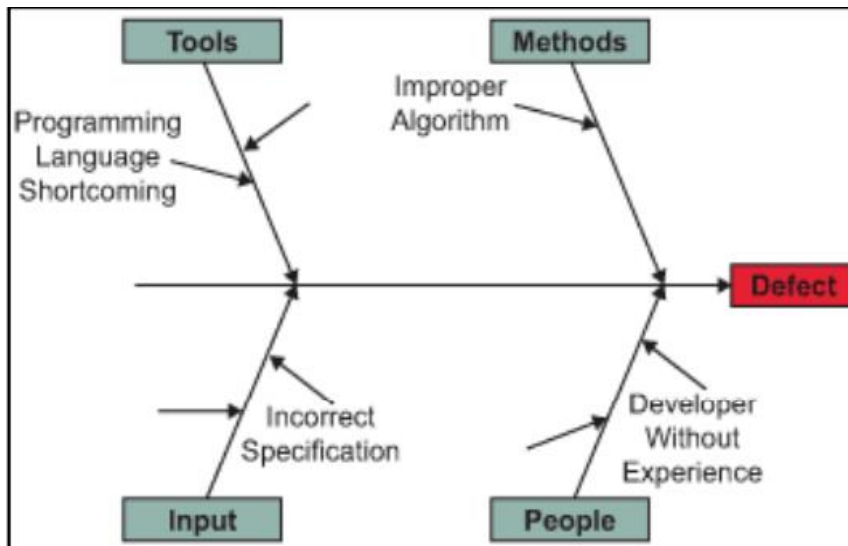
El tercer método de prevención de defectos es consecuencia directa del segundo: con un más exhaustivo, se reduce el tiempo de codificación, la inyección de defectos. Los datos de PSP de 298 ingenieros experimentados muestran el impacto potencial de un buen diseño. Estos datos muestran que, durante el diseño, los ingenieros inyectan una media de 1,76 defectos por hora, mientras que durante la codificación inyectan 4,20 defectos por hora. Dado que se tarda menos tiempo codificar un diseño completamente documentado. Diseño documentado, los ingenieros reducirán de codificación. Por tanto, a los diseños minuciosos, los ingenieros inyectan de codificación (Watts, 2000).



Análisis de la causa raíz, es una forma de encontrar actividades/procesos que causan errores/defectos y también descubre las actividades/procesos para reducir los defectos mediante medidas correctoras. El análisis se basa en dos principios fundamentales:

1. Auto-revisión: El propio desarrollador mejora la calidad del software revisando las etapas fases del SDLC en las que se encontraron defectos/errores.
2. Revisión por pares: El objetivo de la revisión por pares y de la auto revisión es el mismo (eliminar defectos). Es el mismo (es decir, eliminar los defectos). En revisión por pares, se recurre a expertos técnicos para eliminar los defectos.

El diagrama de causas y defectos se utiliza para conocer las causas que generan defectos. También se conoce como diagrama de espina de pescado. Este diagrama suele elaborarse en una sesión de lluvia de ideas por parte del equipo de trabajo. Una vez que las causas son identificadas, los métodos de eliminación requieren otra sesión de lluvia de ideas.



La identificación temprana y la corrección de defectos son esenciales para garantizar la calidad y la fiabilidad del software final. Mediante la aplicación de técnicas como revisiones de código, pruebas de software y análisis estático, los programadores pueden identificar y solucionar problemas antes de que afecten a los usuarios finales. Estos principios no solo contribuyen a la eficiencia en el desarrollo de software, sino que también ayudan a reducir costos y a mejorar la satisfacción del cliente. En última instancia, la adopción de un enfoque proactivo hacia la detección de defectos promueve la creación de aplicaciones más sólidas y confiables, lo que es esencial en el mundo actual de la tecnología.

## Referencias

*Star Journal*, ( Jan-Mar, 2013), *Science, Technology and Arts Research Journal*. [PDF File].

Ram Chillarege, Bhandari, I. S., Chaar, J., & Wong, M.-Y. (1992, December). *Orthogonal Defect Classification - A Concept for In-Process Measurements*. ResearchGate; IEEE (Institute of Electrical and Electronics Engineers).

[https://www.researchgate.net/publication/3187512\\_Orthogonal\\_Defect\\_Classification\\_-\\_A\\_Concept\\_for\\_In-Process\\_Measurements](https://www.researchgate.net/publication/3187512_Orthogonal_Defect_Classification_-_A_Concept_for_In-Process_Measurements)

K.s, J., & R. Vasantha. (2007). *DRE - A Quality Metric for Component based Software Products*. ResearchGate; unknown.

[https://www.researchgate.net/publication/238692605\\_DRE\\_-  
\\_A\\_Quality\\_Metric\\_for\\_Component\\_based\\_Software\\_Products](https://www.researchgate.net/publication/238692605_DRE_-_A_Quality_Metric_for_Component_based_Software_Products)

DEFECT PREVENTION PLANNING. (2023). *Software Project Management in Practice*.

O'Reilly Online Learning. [https://www.oreilly.com/library/view/software-project-  
management/0201737213/0201737213\\_ch05lev1sec2.html](https://www.oreilly.com/library/view/software-project-management/0201737213/0201737213_ch05lev1sec2.html)