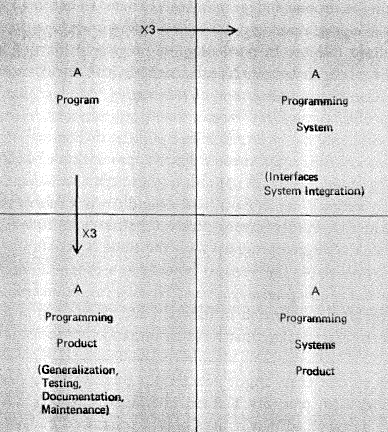
**Producto de los sistemas de programación**

Muchas veces, sucede que dos programadores en un garaje remodelado construyen un importante programa que supera los mejores esfuerzos de grandes equipos de grandes corporaciones. En base a eso, surge la duda de por qué no reemplazar a dichos equipos industriales por dichos programadores.

En el gráfico, se observa un programa, que está completo en sí mismo, listo para ser ejecutado por el autor en el sistema en que fue desarrollado (es decir, de manera local), que es lo que se produce habitualmente en los talleres, y es el objeto que el programador utiliza para estimar la productividad. Hay dos formas de convertir un **programa** en un **objeto**, que esmás útil, pero más costoso (límites del diagrama). Una de ellas, es convertirlo en un **producto de programación**, que es un programa que puede ser ejecutado, reparado y ampliado por cualquiera, puede ser utilizado en entornos operativos, para muchos conjuntos de datos, es decir, es más para uso general. Para convertirse en un producto de programación de uso general, el programa debe ser escrito de forma generalizada, es decir, el rango y forma de las entradas debe generalizarse tanto como sea posible. Luego, el programa debe probarse a fondo para que sea lo suficientemente confiable, utilizando un gran banco de casos de prueba que exploren los rangos de entrada y sus límites. Además, requiere una documentación exhaustiva, para que cualquiera pueda utilizarlo, corregirlo y ampliarlo. Finalmente, podemos decir que un producto de programación cuesta al menos el triple que un programa con la misma función.

Si nos vamos hacia la derecha, un programa se convierte en un **componente de un sistema de programación**, que es un conjunto de programas que interactúan de forma coordinada en su función y disciplinados en su formato del modo que el conjunto constituye una unidad completa para grandes tareas. Para convertirse en un componente del sistema de programación, el programa debe ser escrito de manera que cada entrada y salida se ajuste en sintaxis y semántica con las interfaces definidas con precisión. También debe diseñarse para que utilice sólo una parte de los recursos tales como espacio de memoria, tiempo de procesamiento, etc. Finalmente, debe ser probado exhaustivamente con otros componentes del sistema, en todas las combinaciones previstas, ya que la mayoría de errores surgen con interacciones inesperadas entre componentes. Al igual que en el caso anterior, un componente del sistema de programación cuesta el triple que un programa independiente con la misma función, y el costo puede incrementarse si el sistema tiene muchos componentes.

En la esquina inferior derecha, vemos un **producto de sistemas de programación** que difiere de un simple programa en todos los aspectos antes mencionados, por lo que cuesta 9 veces más, pero es el objeto verdaderamente útil, el producto previsto de la mayoría de los esfuerzos de programación del sistema.