

**INSTITUTO TECNOLÓGICO DE TIJUANA
SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN**



Período: Agosto-Diciembre

Carrera: Ing. en Sistemas Computacionales

Materia: Minería de Datos

Unidad: I Investigación de Pair Coding

No. de control y nombre:

15211916 Vargas García Cinthia Gabriela

Maestro: José Christian Romero Hernández

Fecha de entrega: 16/03/21

Contenido

Pair Programming	3
¿Qué es el pair programming?	3
Prácticas en el pair programming	3
Ventajas y desventajas de la programación en pareja	4
¿Cuáles son los beneficios de este enfoque?	5

Pair Programming

El desarrollo de un software nuevo no es un proceso sencillo. Dependiendo del tamaño del programa, se deberá tener en cuenta una gran cantidad de posibles coyunturas, funciones y cuestiones problemáticas. Incluso los desarrolladores de software más expertos pueden llegar a desorientarse. De ahí que en estos últimos años se hayan ido desarrollando otros métodos de trabajo más modernos que permiten programar con mayor eficiencia y generar un código libre de errores: Scrum y Kanban sirven, por ejemplo, para mejorar el sistema completo.

El pair programming no trata de abarcar tanto: los desarrolladores trabajan siempre por pares en el código. ¿Cómo funciona y cuáles son las ventajas de este método de trabajo?

¿Qué es el pair programming?

El método conocido como pair programming (en español, programación en pareja) se utiliza principalmente en el desarrollo ágil de software y, más concretamente, en la programación extrema (XP). El pair programming especifica que siempre haya dos personas trabajando al mismo tiempo en el código y que, en la medida de lo posible, se sienten juntas. Una se encarga de escribir el código y la otra de supervisarlo en tiempo real. Al mismo tiempo, están constantemente intercambiando impresiones: debaten problemas, encuentran soluciones y desarrollan ideas creativas.

Por lo general, a estos dos trabajadores se les asignan diferentes roles: el programador al que se le ha asignado el rol de piloto se encarga de escribir el código. El programador al que se le ha asignado el rol de copiloto se encarga de supervisar ese código. Una de las reglas del pair programming establece que estos dos roles se intercambien con regularidad (a intervalos cortos). De esta manera se evita una posible brecha jerárquica: se fomenta la igualdad entre ambos trabajadores y se consigue un intercambio fluido de roles.

Además, lo ideal es que el espacio de trabajo también se adapte a los requisitos específicos del pair programming. Cada trabajador deberá tener su propio ratón, teclado y pantalla, en la que se mostrará siempre la misma información que en la del compañero.

Algo menos habitual es el método denominado remote pair programming. En este caso, los programadores no se sientan juntos, si no que están ubicados en lugares completamente diferentes. Para que este método funcione, se debe contar con soluciones técnicas especiales. Aun a pesar de la distancia, los compañeros deben tener una línea de comunicación directa y deben poder acceder al código y visualizar las modificaciones en tiempo real.

Prácticas en el pair programming

En la práctica, esta suele ser una colaboración entre dos desarrolladores con diferentes grados de experiencia: así, un trabajador con más experiencia puede transmitir sus

conocimientos a sus compañeros más jóvenes directamente a través de la práctica. Por su parte, es posible que a un trabajador más joven se le ocurran otras ideas, quizá más innovadoras, que puede aportar al proyecto.

Asimismo, una combinación de trabajadores de diferentes sectores también puede resultar bastante beneficiosa: si un programador de la vieja escuela colabora con un diseñador, la combinación de sus experiencias puede ser de bastante ayuda tanto para uno como para otro.

El método de trabajo resulta de gran utilidad principalmente para proyectos grandes. El principio de “los cuatro ojos” es especialmente eficaz cuando se tiene que trabajar con grandes cantidades de código que, además, debe modificarse con regularidad. Garantiza que en el texto de origen se va a introducir siempre la mejor versión posible de un fragmento, reduciéndose la cantidad de errores, de modo que no será necesario realizar tantos retoques *a posteriori*. La supervisión posterior de códigos fuente muy largos requiere invertir demasiado tiempo y esfuerzo, por lo que lo más conveniente es que estos códigos se establezcan sin errores desde el principio en la medida de lo posible.

Sin embargo, no siempre es obligatorio que esta colaboración sea entre los mismos compañeros, incluso en el caso de un mismo proyecto. De hecho, puede resultar beneficioso mezclar con regularidad estas parejas. De este modo, todos los miembros del equipo podrán tener una buena idea del texto de origen completo. Esto permite que el éxito del proyecto no dependa tanto de las diferentes personas involucradas. Si una de ellas falla, no pone en peligro todo el proyecto, pues el resto puede compensar su ausencia.

Ventajas y desventajas de la programación en pareja

El trabajo en pareja en el marco de un proyecto, tanto si se trata de la programación como de otro procedimiento, lleva consigo gran cantidad de ventajas. Por lo general, cuatro ojos ven mucho más que solo dos: con el método pair programming se minimiza el riesgo de que se produzcan errores. Mientras una persona escribe el código, la otra lo visualiza y se concentra tan solo en la búsqueda de errores. Por lo general, nos resulta bastante complicado detectar los propios errores. Suelen ser nuestros compañeros quienes los detectan con más rapidez.

Otra de las grandes ventajas derivadas de la comunicación es el desarrollo constante de la creatividad: el constante intercambio que se produce entre la pareja de programadores hace que surjan ideas que quizá no se tendrían si el trabajo fuese individual. La intercomunicación también garantiza que los problemas se puedan solucionar mejor en menos tiempo. Pues, mientras una persona que trabaja sola puede sentirse satisfecha con la primera opción que mejor le parezca, las personas involucradas en el pair programming siempre deben justificar sus decisiones ante el resto. Es posible que estas tengan otra percepción del problema y no estén satisfechas con la solución que se les propone. Esto genera un debate en el que suelen presentarse nuevas ideas que derivan en un código mucho mejor.

Un buen código también es un código conciso: la experiencia dicta que un texto de origen generado por pair programming suele tener un diseño más corto y, por tanto, más eficiente. Esto permite un ahorro posterior en recursos en caso de mantenimiento y adaptación.

Como ya hemos mencionado, esta técnica también puede aprovecharse para que los trabajadores con más experiencia compartan sus conocimientos con sus compañeros más jóvenes. De este modo, no solo se obtiene beneficio de la ventaja esencial del pair programming, que es la generación de un código de alta calidad, sino que también se puede utilizar simultáneamente para fines de formación.

Sin embargo, todo esto requiere mucho tiempo: dos programadores juntos trabajan mucho más rápido que uno solo, pero no más que dos programadores que trabajan por separado. Esto quiere decir que este método hace que los proyectos avancen más lentamente o que requieran más personal, lo cual aumenta a su vez los costes. Los partidarios del pair programming estiman que estas horas extraordinarias se compensan pues el código generado contiene menos errores, está mejor estructurado en general y requiere mucho menos mantenimiento.

Otra posible desventaja es que el pair programming es adecuado para la formación de equipos, pero solo si ambos compañeros trabajan bien juntos. Al tratarse de una colaboración tan estrecha, es posible que los problemas que puedan tener entre ellos ralenticen los resultados o acaben aumentando hasta algo mucho peor. Es por este motivo que la asignación de parejas en este método no puede realizarse aleatoriamente. Lo ideal sería trabajar cada vez con un compañero diferente, pero esto solamente funciona bien si hay armonía en todo el equipo

¿Cuáles son los beneficios de este enfoque?

Mayor disciplina: La presencia del observador impide que el driver se distraiga de sus tareas y lo obliga a hacer las cosas bien.

Mayor motivación: La programación en pares es más disfrutable que la programación en solitario, siendo una importante motivación para el equipo de trabajo.

Mayor productividad a largo plazo: De acuerdo con un estudio, pair programming aumenta en un 15% el tiempo de desarrollo, pero a cambio brinda más calidad en el diseño, reduce los defectos y mejora la comunicación entre los miembros del equipo.

Mejor aprendizaje: El driver puede ser acompañado por un observador más experimentado que él, o viceversa. Incluso es probable que tengan un nivel de habilidad similar en distintos campos. Como resultado, al menos uno de los dos aprenderá de la mejor manera: viendo a un experto trabajar.

Reducción de riesgos: Como el código estará sometido a una revisión constante, se reducirá el riesgo de cometer errores que nos alejen de la solución deseada.

Bibliografía

1&1 IONOS Inc. (2020, 30 septiembre). Pair Programming. IONOS Digital Guide.
<https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/pair-programming/>

A. (2016, 3 noviembre). *Pair programming: mejorando el proceso de desarrollo*.
4R Soluciones | Diseño, Desarrollo y Programación Web & Mobile.
<https://www.4rsoluciones.com/blog/pair-programming-mejorando-el-proceso-de-desarrollo-2/>