

Resumen Capítulo 4 the Clean coder Codificando

En este capítulo analizaremos el acto de codificar, y el contexto que lo rodea.

Ser capaz de detectar sus errores es realmente importante. No sólo escribiendo, sino en todo. Tener error-sentido significa que usted cerrará rápidamente el bucle de retroalimentación y aprender de sus errores con mayor rapidez. He estudiado y dominado varias disciplinas desde ese día en el 029. He descubierto que en cada caso la clave del dominio es la confianza y el error-sentido.

Describiremos el conjunto personal de reglas personales y principios para la codificación. No tratan sobre mi punto de vista subjetivo, son sobre el comportamiento, humor actitud a la hora de escribir código. Estas condiciones definen el contexto mental moral y emocional para codificar. Es así como se arraigan las raíces de la confianza y el error-sentido.

Es probable que no esté de acuerdo con todo lo que digo aquí. Después de todo, esto es profundamente personal cosas. De hecho, usted puede estar en desacuerdo con algunas de mis actitudes y principios. Eso está bien, no tienen la intención de ser verdades absolutas para nadie más que para mí. Lo que son es el enfoque de un hombre para ser un codificador profesional. Tal vez, al estudiar y contemplar mi propio entorno de codificación personal, Puede aprender a arrebatarme el guijarro de mi mano.

PREPARACIÓN

La codificación es una actividad intelectualmente desafiante y agotadora. Requiere un nivel De concentración y enfoque que otras pocas disciplinas requieren. La razón de Esto es que la codificación requiere que usted haga malabares con muchos factores competentes a la vez.

1. Primero, su código debe funcionar. Debes entender qué problema tienes que resolver y entender cómo resolver ese problema. Usted debe asegurarse de que el Código que usted escribe es una representación fiel de esa solución. Debe gestionar todos los detalles de esa solución sin dejar de ser coherente dentro del lenguaje, Plataforma, arquitectura actual, y todas las características del sistema actual.
2. Su código debe resolver el problema establecido por el cliente. A menudo Los requisitos del cliente no solucionan realmente los problemas del cliente. Es usted el señalado para ver esto y debe negociar con el cliente para asegurarse de que las verdaderas necesidades del cliente son conocidas.
3. Su código debe encajar bien en el sistema existente. No debería aumentar la Rigidez, fragilidad u opacidad de ese sistema. Las dependencias deben estar bien administradas. En resumen, su código debe seguir principios sólidos de ingeniería.
4. Su código debe ser legible por otros programadores. Esto no es simplemente un Cuestión de escribir buenos comentarios. Por el contrario, requiere que elaborar el

código De tal manera que revele su intención. Esto es difícil de hacer. De hecho, esto puede ser La cosa más difícil que un programador puede dominar.

Jugar con todos estos asuntos es difícil. Es fisiológicamente difícil mantener el La concentración necesaria y el enfoque durante largos períodos de tiempo. Añade a esto los Problemas y distracciones de trabajar en equipo, en una organización y Cuidados e inquietudes de la vida cotidiana. La conclusión es que la oportunidad para La distracción es alta. Cuando no puede concentrarse y enfocarse lo suficiente, el código que escriba será incorrecto. Tendrá errores. Tendrá la estructura equivocada. Será opaco y complejo. No resolverá los problemas reales de los clientes. En resumen, tendrá que ser reelaborado o rehecho. Trabajar mientras se distrae crea desperdicio. Si está cansado o distraído, no codifique. Sólo terminará rehaciendo lo que hizo. En su lugar, encuentre una manera de eliminar las distracciones y calmar su mente.

CÓDIGO DE LAS 3 DE LA MAÑANA

El peor código que he escrito fue a las 3 am. El año era 1988, y yo estaba trabajando En una empresa de telecomunicaciones llamada Clear Communications. Todos estábamos poniendo largas horas de trabajo bajo el principio de "la equidad del sudor." Estábamos, por supuesto, todos muy tarde, o mejor mas bien temprano, resolviendo un problema del código que debía enviar un mensaje por medio del evento despacho del sistema llamado como "enviar un correo electronico". Esto fue una solución equivocada. Pero no éramos conscientes, y se veía bien a esa hora, de hecho después de 16 horas de codificación diaras era todo lo que se podía pensar.

Recuerdo sentirme tan bien conmigo mismo durante las largas horas que estuve trabajando. Recuerdo haberme sentido dedicado. Recuerdo pensar que trabajar a las 3 am es lo que Profesionales serios ahcen. ¡Qué equivocado estaba!

Ese código volvió a mordernos una y otra vez. Instituyó un diseño estructural defectuoso que todo mundo usaba pero que consistentemente debían trabajar alrededor de él. Causó todo tipo de extraños errores de medición del tiempo y bucles de retroalimentación. Nos metimos en el infinito ciclo de correos donde un mensaje causaba que otro se enviara, y después otro, infinitamente. Nunca tuvimos tiempo de corregir este error, pero siempre pensábamos que ahbria tiempo para agregar otro parche para trabajar sobre el. El problema crecía y crecía, bordeando ese código de 3 am con mas cosas y efectos laterales. Cuando estábamos cansados o frustados lo recordábamos con jocosidad.

La moraleja de esta historia es no escribir código cuando esté cansado. Para alcanzar dedicación y profesionalismo se necestia mas disciplina que horas. Asegúrese de dormir, de su salud y un estilo de vida donde enfunde 8 buenas horas laborales.

Código de preocupación

¿Alguna vez se ha metido en una pelea grande con su cónyuge o amigo, y luego intentó ¿codificar? ¿Se dio cuenta de que había un proceso de fondo en ejecución en su Mente tratando de resolver, o al menos revisar la pelea? A veces puedes sentir la Estrés de ese proceso de fondo en su pecho, o en el pozo de su estómago. Puede hacerte sentir ansioso, como cuando has tomado demasiado café o dieta Coca.

Cuando estoy preocupado por una discusión con mi esposa, o una crisis de clientes, o una Niño enfermo, no puedo mantener el enfoque. Mi concentración oscila. Me encuentro con Mis ojos en la pantalla y mis dedos en el teclado, sin hacer nada. Catatónico. Paralizado. Totalmente alejado de la codificación. En ocasiones uno se obliga a codificar, a realizar pruebas. Pero no se puede seguir así. Inevitablemente desciende uno a una insensibilidad estupefacta, donde no se ve nada con los ojos abiertos, agitando la preocupación en el interior. Ese no es tiempo para codificar. Cualquier código que produzca será basura. Por tanto es mejor resolver la preocupación.

Por supuesto, hay muchas preocupaciones que simplemente no se pueden resolver en una hora o dos. Además, es poco probable que nuestros empleadores toleren que trabajemos a medida que resolvemos nuestros problemas personales. El truco es aprender a cerrar Proceso de fondo, o al menos reducir su prioridad para que no sea una Distracción continua.

Hago esto dividiendo mi tiempo. En lugar de obligarme a codificar mientras La preocupación de fondo es irritante para mí, voy a pasar un bloque dedicado de tiempo, Tal vez una hora, trabajando en el tema que está creando la preocupación. Si mi hijo es Enfermo, voy a llamar a casa y el check in. Si he tenido una discusión con mi esposa, voy a llamar Ella y hablar a través de los temas. Si tengo problemas con el dinero, pasaré tiempo Pensando en cómo puedo lidiar con los problemas financieros. Sé que no es probable que Resolver los problemas en esta hora, pero es muy probable que pueda reducir la ansiedad Y silenciar el proceso de fondo.

Lo ideal es que el tiempo gastado en la lucha con los problemas personales debería ser en el tiempo personal. Eso sería una pena pasar una hora en la oficina de esta manera. Desarrolladores profesionales Asignan su tiempo personal para garantizar que el tiempo empleado en la oficina sea Tan productivo como sea posible. Eso significa que usted debería reservar tiempo Casa para resolver sus ansiedades para que no los traiga a la oficina.

Por otro lado, si usted se encuentra en la oficina y las ansiedades en el fondo están socavando su productividad, entonces es mejor pasar una hora tranquilizándose que usar la fuerza bruta para escribir código que solo tendrás que Tirar más tarde (o peor, vivir con el).

LA ZONA DE FLUJO

Mucho se ha escrito sobre el estado hiper-productivo conocido como "flujo". Algunos programadores lo llaman "la Zona". Lo que se llame, probablemente Familiarizado con él. Es el estado altamente enfocado de visión de túnel de la conciencia Que los programadores pueden entrar mientras escriben código. En este estado se sienten productivo. En este estado se sienten infalibles. Y así ellos desean alcanzar eso Estado, ya menudo miden su autoestima por cuánto tiempo pueden Pasar allí

Aquí hay una pequeña pista de alguien que ha estado allí y atrás: Evite la zona. Este estado de conciencia no es realmente hiperproductivo y ciertamente no es infalible. En realidad, es sólo un estado meditativo suave en el que ciertas facultades racionales disminuyen a favor de una sensación de velocidad.

Permítanme ser claro acerca de esto. Escribiré más código en la zona. Si usted es Practicando TDD, recorrerás el bucle rojo / verde / refactor más rápidamente. Y sentirá una leve euforia o un sentido de conquista. El problema es ese Usted pierde algo de la imagen grande mientras está en la Zona, por lo que probablemente Tomar decisiones que más tarde tendrá que retroceder y retroceder. Código escrito en La zona puede salir más rápido, pero volverá a visitarlo más.

Uno de los grandes beneficios de la programación de pares es que es prácticamente imposible Un par para entrar en la Zona. La Zona es un estado poco comunicativo, mientras que el emparejamiento Requiere una comunicación intensa y constante. De hecho, una de las quejas que ha menudo se oye acerca de emparejamiento es que bloquea la entrada en la zona. ¡Bueno! La zona No es donde quieres estar. Bueno, eso no es verdad. Hay veces en que la Zona es exactamente donde usted quiero ser. Cuando usted está practicando.

MUSICA

Bueno, te haces una idea.

Solía poner el estéreo en marcha y luego escribir código. Pensé que ayudó a mi concentración. Pero estaba equivocado.

Ahí es cuando me golpeó. Como lector del código, estaba aprendiendo más sobre el Colección de música del autor (yo) que yo estaba aprendiendo sobre el problema que El código estaba tratando de resolver.

Me di cuenta de que simplemente no código bien mientras escucha música. La musica no No me ayuda a concentrarme. De hecho, el acto de escuchar música parece consumir Recurso vital que mi mente necesita para escribir código limpio y bien diseñado.

Quizás no funciona de esa manera para usted. Tal vez la música le ayuda a escribir código. yo Conocer a mucha gente que codifica mientras lleva auriculares. Acepto que la música Puede ayudarlos, pero también sospecho que lo que realmente está sucediendo es que la La música les está ayudando a entrar en la Zona.

INTERRUPCIONES

Visualízate como estás codificando en tu estación de trabajo. Como respondes Cuando alguien te hace una pregunta? ¿Usted chasquea en ellos? ¿Eres fulminante? Tu ¿El lenguaje corporal les dice que se vayan porque están ocupados? En resumen, ¿eres grosero?

O, ¿dejas de hacer lo que estás haciendo y amablemente ayudas a alguien que está atrapado? Los tratas como si quisieras que te trataran si estuvieras atascado? La respuesta grosera a menudo viene de la Zona. Es posible que te moleste que te arrastren Fuera de la Zona, o puede que le moleste que alguien interfiera con su intento de Entrar en la Zona. De cualquier manera, la grosería a menudo viene de su relación con la zona. A veces, sin embargo, no es la zona que tiene la culpa, es sólo que usted está tratando DE entender algo complicado que requiere concentración. Existen Varias soluciones a esto.

El emparejamiento puede ser muy útil como una forma de lidiar con las interrupciones. Tu pareja Puede sostener el contexto del problema en la mano, mientras que usted trata de una llamada de teléfono, O una pregunta de un compañero de trabajo. Cuando regresa a su pareja, rápidamente Le ayuda a reconstruir el contexto mental que tenía antes de la interrupción. TDD es otra gran ayuda. Si usted tiene una prueba fallida, esa prueba tiene el contexto de Dónde estás. Puede volver a él después de una interrupción y continuar Que falla el pase de prueba. Al final, por supuesto, habrá interrupciones que te distraigan y te causen Perder tiempo Cuando sucedan, recuerde que la próxima vez usted puede ser el Que necesita interrumpir a alguien más. Así que la actitud profesional es un educado Disposición a ser útil.

BLOQUEO PARA CODIFICAR

A veces el código simplemente no aparece. He tenido esto me sucede y he visto Sucede a otros. Te sientas en tu puesto de trabajo y no pasa nada. A menudo encontrará otro trabajo que hacer. Leerás correo electrónico. Usted leerá tweets. Usted Mirar a través de libros, o planificaciones, o documentos. Llamaré a reuniones. Usted Iniciar conversaciones con otros. Harás cualquier cosa para que no tengas que Hacer frente a esa estación de trabajo y ver como el código se niega a aparecer.

¿Qué causa tales bloqueos? Hemos hablado de muchos de los factores ya. Para mí, otro factor importante es el sueño. Si no estoy durmiendo lo suficiente, simplemente No puede codificar. Otros son la preocupación, el miedo y la depresión. Curiosamente hay una solución muy simple. Funciona casi cada vez. Es fácil Para hacer, y puede proporcionarle el impulso para conseguir porciones de código escrito. La solución: Encuentra un par de parejas. Es extraño lo bien que esto funciona. Tan pronto como te sientas junto a otra persona, Los problemas que te estaban bloqueando se derriten. Hay un cambio fisiológico que Ocurre cuando se trabaja con alguien. No sé lo que es, pero puedo Definitivamente lo siento. Hay algún tipo de cambio químico en mi cerebro o cuerpo que Me rompe a través del bloqueo y me vuelve a ir. Esta no es una solución perfecta. A veces el cambio dura una hora o dos, sólo Para ser seguido por el agotamiento tan grave que tengo que romper con mi pareja Encontrar algún agujero para recuperarse. Algunas veces, incluso cuando Alguien, no puedo hacer más que simplemente de acuerdo con lo que esa persona está haciendo. Pero para La reacción típica al emparejamiento es una recuperación de mi impulso.

ENTRADAS CREATIVAS

Hay otras cosas que hago para evitar el bloqueo. Aprendí hace mucho tiempo que La salida creativa depende de la entrada creativa. Leo mucho y leo todo tipo de material. He leído material sobre software, política, Biología, astronomía, física, química, matemáticas y mucho más. Sin embargo, Encuentro que la cosa que mejor prime la bomba de la salida creativa es ciencia ficción. Para ti, podría ser otra cosa. Tal vez una buena novela de misterio, o poesía, o Incluso una novela romántica. Creo que el verdadero problema es que la creatividad genera creatividad. También hay un elemento de escapismo. Las horas que paso lejos de mi habitual Problemas, al mismo tiempo que se estimulan activamente mediante ideas desafiantes y creativas, Resulta en una presión casi irresistible para crear algo yo mismo.

No todas las formas de trabajo creativo de entrada para mí. Ver televisión no suele ayudar Yo creo Ir al cine es mejor, pero sólo un poco. Escuchar música no No me ayuda a crear código, pero sí me ayuda a crear presentaciones, charlas y Videos. De todas las formas de entrada creativa, nada funciona mejor para mí que Buena vieja ópera espacial.

Por alguna razón los desarrolladores de software no piensan en depurar el tiempo como codificación hora. Piensan en depurar el tiempo como una llamada de la naturaleza, algo que acaba de para acabar. Pero el tiempo de depuración es tan caro para el negocio como la codificación El tiempo es, y por lo tanto cualquier cosa que podemos hacer para evitarlo o disminuirlo es bueno. Hoy en día paso mucho menos tiempo depurando que hace diez años. No tengo Midió la diferencia, pero creo que se

trata de un factor de diez. Logré esto Una reducción verdaderamente radical en el tiempo de depuración adoptando la práctica de Test Driven Development (TDD), que discutiremos en otro capítulo. Si usted adopta TDD o alguna otra disciplina de igual eficacia, 3 es Incumbe a usted como un profesional para reducir su tiempo de depuración tan cerca A cero como usted puede conseguir. Claramente cero es un objetivo asintótico, pero es la meta sin embargo. A los médicos no les gusta reabrir a los pacientes para arreglar algo que hicieron mal. Abogados No les gusta volver a intentar los casos que flubbed. Un médico o abogado que hizo eso Demasiado a menudo no sería considerado profesional. Asimismo, un desarrollador de software Que crea muchos errores está actuando de forma no profesional.

PONGASE EN CONCTACTO CON USTED

El desarrollo de software es un maratón, no un sprint. No puedes ganar la carrera por Tratando de correr tan rápido como puedas desde el principio. Usted gana conservando su Recursos y ritmo usted mismo. Un corredor de maratón se encarga de su cuerpo tanto Antes y durante la carrera. Los programadores profesionales conservan su energía y Creatividad con el mismo cuidado.

SABER CUANDO IRSE

¿No puedes ir a casa hasta que resuelvas este problema? Oh sí puedes, y probablemente ¡debería! La creatividad y la inteligencia son estados fugaces de la mente. Cuando estás Cansados, se van. Si luego golpea su cerebro que no funciona durante horas después Hora de la noche tratando de resolver un problema, simplemente te haces más Cansado y reducir la posibilidad de que la ducha, o el coche, le ayudará a resolver el problema. Cuando estás atascado, cuando estás cansado, desactiva por un tiempo. Entrega tu Subconsciente creativo una grieta en el problema. Usted conseguirá más hecho en menos Tiempo y con menos esfuerzo si usted tiene cuidado de esposar sus recursos. Paso Usted y su equipo. Aprenda sus patrones de creatividad y brillantez, y Tomar ventaja de ellos en lugar de trabajar en contra de ellos.

MANEJANDO A CASA

Un lugar que he resuelto una serie de problemas es mi coche en el camino a casa Del trabajo. Conducir requiere muchos recursos mentales no creativos. Debes Dedicar tus ojos, manos y porciones de tu mente a la tarea; Por lo tanto, usted Debe desconectarse de los problemas en el trabajo. Hay algo Que permite que su mente busque soluciones de una manera diferente y Más creativa.

EL BAÑO

He resuelto un número excesivo de problemas en la ducha. Quizás eso Spray de agua temprano en la mañana me despierta y me lleva a revisar todos los Soluciones que mi cerebro se le ocurrió mientras yo estaba dormido. Cuando estás trabajando en un problema, a veces te acercas tanto que No puede ver todas las opciones. Te pierdes las soluciones elegantes porque la parte creativa de Su mente se suprime por la intensidad de su enfoque. A veces la mejor manera Para resolver un problema es ir a casa, comer la cena, ver la televisión, ir a la cama, y luego Despertar a la mañana siguiente y tomar una ducha. Abajo

LLEGAR TARDE

Llegaras tarde. Le pasa a los mejores de nosotros. Sucede al más dedicado de nos. A veces simplemente soplamos nuestras estimaciones y terminamos tarde. El truco para manejar el retraso es la detección temprana y la transparencia. Lo peor Caso ocurre cuando usted continúa a decir a todos, hasta el final, Que llegará a tiempo, y luego los dejará a todos. No lo hagas. En lugar, Regularmente medir su progreso en contra de su meta, y llegar a tres⁴ Fechas de fin de hechos: mejor caso, caso nominal y peor caso. Sea tan honesto como Usted puede sobre las tres fechas. ¡No incorpore la esperanza en sus estimaciones! Presente los tres números a su equipo ya las partes interesadas. Actualizar estos Números diarios.

ESPERANZA

¡No espere que usted consiga todo hecho en diez días! La esperanza es el asesino del proyecto. La esperanza destruye los horarios y arruina la reputación. La esperanza te lleva a lo profundo Problema Si la feria es en diez días, y su estimación nominal 12, usted No van a hacerlo. Asegúrese de que el equipo y las partes interesadas Entender la situación, y no dejar hasta que haya un plan de repliegue. No Que no hay esperanza

CORRIENDO

¿Qué pasa si su gerente se sienta y le pide que trate de hacer la fecha límite?
¿Qué pasa si su gerente insiste en que "hacer lo que se necesita"? Mantenga a sus estimaciones!
Sus estimaciones originales son más exactas que cualquier cambio que realice mientras ..

Tu jefe te está enfrentando. Dile a tu jefe que ya has considerado el Opciones (porque usted tiene) y que la única manera de mejorar el horario es Reducir el alcance. No se sienta tentado a precipitarse.
¡Ay del pobre desarrollador que se dobla bajo presión y acepta intentar Hacer el plazo. Ese desarrollador comenzará a tomar atajos y trabajar extra Horas en la vana esperanza de hacer un milagro. Esta es una receta para el desastre porque Le da a usted, a su equipo ya sus interesados una falsa esperanza. Permite a todos Evitar enfrentar el problema y retrasar las decisiones difíciles necesarias.

No hay manera de precipitarse. No se puede hacer el código más rápido. No puedes hacer Resolver los problemas más rápido. Si lo intentas, solo te ralentizarás y Hacer un lío que se ralentiza todo el mundo hacia abajo, también. Así que usted debe responder a su jefe, su equipo, y sus accionistas privando Ellos de esperanza.

HORAS EXTRA

Así que su jefe dice, "¿Qué pasa si usted trabaja un extra de dos horas al día? Y si trabajas ¿en sábado? Vamos, sólo tiene que ser una manera de exprimir suficientes horas en Para obtener la característica a tiempo. " Las horas extras pueden funcionar, ya veces es necesario. A veces puedes hacer una Fecha imposible de poner en unos diez horas diarias, y un sábado o dos. Pero esto es muy arriesgado. No es probable que obtenga un 20% más de trabajo realizado por Trabajando un 20% más de horas. Es más, las horas extraordinarias sin duda fallarán si continúa Por más de dos o tres semanas. Por lo tanto, no debe aceptar trabajar horas extras a menos que (1) pueda personalmente (2) es de corto plazo, dos semanas o menos, y (3) su jefe tiene un retroceso Plan en caso de que el esfuerzo de horas extras falla. Ese último criterio es un trato de acuerdo. Si su jefe no puede articular a usted lo que Él va a hacer si el

esfuerzo de tiempo extra falla, entonces usted no debe aceptar trabajar a través del tiempo.

FALSA ENTREGA

De todos los comportamientos poco profesionales que un programador puede disfrutar, tal vez lo peor de todo es decir que se hacen cuando saben que no lo son. A veces esto es sólo una mentira abierta, y eso es bastante malo. Pero el caso más insidioso es cuando logramos racionalizar una nueva definición de "hecho". Nosotros mismos que hemos hecho lo suficiente, y pasar a la siguiente tarea. Racionalizamos que cualquier trabajo que queda puede ser tratado más tarde cuando tengamos más tiempo. Esta es una práctica contagiosa. Si un programador lo hace, otros lo verán y seguir el ejemplo. Uno de ellos extenderá aún más la definición de "hecho", y todos los demás adoptarán la nueva definición. He visto esto llevado a horribles extremos. Uno de mis clientes realmente definió "hecho" como "registrado". El código ni siquiera tenía que compilar. ¡Es muy fácil ser "hecho" si nada tiene que trabajar! Cuando un equipo cae en esta trampa, los gerentes oyen que todo va bien. Todos los informes de estado muestran que todos están a tiempo. Es como hombres ciegos que tienen un picnic en las vías del ferrocarril: Nadie ve el tren de carga de trabajo sin terminar hasta que sea demasiado tarde.

DEFINICION DE LISTO

Evita el problema de la entrega falsa creando una definición independiente de "Hecho". La mejor manera de hacer esto es tener a sus analistas de negocio y probadores crear pruebas de aceptación automatizadas que deben pasar antes de poder decir que usted están hechos. Estas pruebas deben ser escritas en un lenguaje de prueba como FitNesse, Selenium, RobotFX, pepino, y así sucesivamente. Las pruebas deben ser comprensibles por las partes interesadas y la gente de negocios, y debe ser ejecutado con frecuencia.

AYUDA

La programación es difícil. Cuanto más joven eres, menos crees esto. Después de todo, es sólo un montón de declaraciones if y while. Pero a medida que adquiere experiencia, empieza a darse cuenta de que la forma en que se combinan las declaraciones if y while es críticamente importante. Usted no puede simplemente slather juntos y esperar lo mejor. Más bien, usted tiene que cuidadosamente dividir el sistema en pequeñas unidades comprensibles que tienen tan poco que ver con los demás como sea posible, y eso es difícil. La programación es tan difícil, de hecho, que está más allá de la capacidad de una persona para hacerlo bien. Independientemente de su habilidad, usted se beneficiará de pensamientos e ideas de otro programador.

Debido a esto, es la responsabilidad de los programadores estar disponibles para ayudar el otro al otro. Es una violación de la ética profesional secuestrarse en una cubículo u oficina y rechazar las consultas de otros. Tu trabajo no es tan importante que no puede prestar algo de su tiempo para ayudar a otros. De hecho, como profesional usted es honor obligado por ofrecer esa ayuda siempre que sea necesario. Esto no significa que usted no necesita un poco de tiempo solo. Por supuesto que sí. Pero usted tiene que ser justo y educado sobre él. Por ejemplo, puedes dejarlo saber que entre las horas de las 10 am y el mediodía no debe ser molestado, pero de 1 pm a 3 pm su puerta está abierta.

Usted debe ser consciente del estado de sus compañeros de equipo. Si ves a alguien Que parece estar en problemas, usted debe ofrecer su ayuda. Probablemente será bastante Sorprendido por el profundo efecto que su ayuda puede tener. No es que seas tan Mucho más inteligente que la otra persona, es sólo que una nueva perspectiva puede Profundo catalizador para resolver problemas. Cuando ayudas a alguien, siéntate y escribe código juntos. Plan para pasar el Mejor parte de una hora o más. Puede tomar menos que eso, pero usted no quiere Parecen ser apresurado. Resignarse a la tarea y darle un esfuerzo sólido. Tú Probablemente saldrá aprendiendo más de lo que dio.

SIENDO AYUDADO

Cuando alguien se ofrece para ayudarle, sea amable con él. Acepta la ayuda Con gratitud y darse a esa ayuda. No proteja su césped. No empuje

La ayuda de distancia porque usted está bajo el arma. Déle treinta minutos más o menos. Si por Esa vez la persona realmente no está ayudando tanto, entonces excusa cortésmente Y termina la sesión con agradecimiento. Recuerda, tal como eres Honor obligado a ofrecer ayuda, que están obligados a aceptar la ayuda. Aprenda a pedir ayuda. Cuando usted está atascado, o befuddled, o simplemente no puede envolver Tu mente alrededor de un problema, pide ayuda a alguien. Si usted está sentado en un equipo , Puede sentarse y decir: "Necesito ayuda". De lo contrario, use yammer, O twitter, o correo electrónico, o el teléfono en su escritorio. Llamar por ayuda. De nuevo, este es un Cuestión de la ética profesional. Es poco profesional permanecer atascado cuando la ayuda es De fácil acceso.

Para este momento usted puede estar esperando que yo estalle en un estribillo de Kumbaya mientras que Los conejitos borrosos saltan sobre las partes posteriores de los unicornios y todos volamos alegremente Arco iris de esperanza y cambio. No, no del todo. Usted ve, los programadores tienden a ser Introvertidos arrogantes y egoístas. No entramos en este negocio porque como personas. La mayoría de nosotros entró en la programación porque preferimos enfocar profundamente

En minucias estériles, combinan muchos conceptos simultáneamente y, en general, prueban A nosotros mismos que tenemos cerebros del tamaño de un planeta, todos sin tener que Interactuar con las complicadas complejidades de otras personas.

Sí, esto es un estereotipo. Sí, es una generalización con muchas excepciones. Pero el La realidad es que los programadores no tienden a ser colaboradores.⁶ Sin embargo, la colaboración Es fundamental para la programación eficaz. Por lo tanto, ya que para muchos de nosotros la colaboración No es un instinto, requerimos disciplinas que nos impulsen a colaborar.

MENTORING

Tengo todo un capítulo sobre este tema más adelante en el libro. Por ahora déjeme decir simplemente Que la formación de programadores menos experimentados es responsabilidad de los Que tienen más experiencia. Los cursos de formación no lo cortan. Los libros no lo cortan. Nada puede llevar a un desarrollador de software joven de alto rendimiento más rápido

Que su propio impulso, y la tutoría eficaz por sus mayores. Por lo tanto, una vez Una vez más, es una cuestión de ética profesional para los programadores senior pasar tiempo Teniendo programadores más jóvenes bajo su ala y mentorándolos. Por el Los programadores más jóvenes tienen el deber profesional de buscar Tales mentores de sus seniors.r