

estás familiarizado quizás con las metodologías ágiles pero el nombre de Extreme programming o XP para abreviar sigue siendo un misterio para ti suena bueno Interesante pero extremo verdad quizás no estás seguro de que esta metodología sea para ti sin embargo no dejes que este nombre te distraiga porque te perderías de muchas cosas buenas en Este vídeo lo que pretendo explicarte es todo lo que necesitas saber sobre la programación extrema o Stream programming para que también puedas usarla a tu favor soy Gina Ruiz Y si te interesa de lo que vamos a hablar en el día de hoy Quédate hasta el final del vídeo que enseguida comenzamos iniciemos por definir Qué es la programación extrema XP o Extreme programming pues la programación extrema es una metodología de desarrollo de Software que forma parte de lo que colectivamente se conoce como metodologías ágiles XP se basa en valores principios y práctica y su objetivo es permitir que los equipos pequeños y medianos produzcan software de alta calidad y se adapten a los requisitos y evolución del cambio lo que diferencia a XP de las otras metodologías ágiles es que XP hace hincapié en los aspectos técnicos del desarrollo de software la programación extrema es precisa sobre Cómo trabajan los ingenieros ya que seguir las prácticas de ingeniería permite a los equipos entregar código de alta calidad a un ritmo sostenible la programación extrema es en pocas palabras sobre las buenas prácticas llevadas al extremo dado que la programación en pares es buena Pues hagámoslo todo el tiempo y dado que probar es bueno probemos antes de que se escriba el código de producción Pero antes de entrar en detalles sobre Qué es la metodología de gestión de proyectos ágil Extreme programming hablemos un poco de su historia El origen de XP se remonta a los años 90 Cuando Kent Beck se convertiría en uno de los autores del manifiesto ágil

lo creó cuando fue contratado para dirigir el equipo del sistema de compensación integral de Christle. El proyecto se creó comenzando en el 1993 y en el 1996 no había progresado mucho dado que Beck era nuevo en la gestión de un equipo decidió que el mejor curso de acción sería enseñar a los miembros de su equipo las técnicas y prácticas. Que funcionaron para él comenzaron a aplicar prácticas como la programación por pares y tbd con gran éxito.

Ron Jeffries un amigo de Beck y otro autor del manifiesto ágil fue traído para entrenar al equipo de C3 en 1999 y luego Ken Baker formalizó las prácticas principios y valores de XP en su libro Xtreme programming en xplanet.

Ahora sí expliquemos cómo funciona la programación extrema.

XP A diferencia de otras metodologías es muy obstinado cuando se trata de prácticas de ingeniería además de las prácticas XP se basa en valores y principios. Los valores proporcionan un propósito a los equipos actúan como una estrella del Norte para guiar las decisiones de alto nivel sin embargo los valores son abstractos y demasiados borrosos para una orientación específica. Por ejemplo decir que valoras la comunicación puede resultar en muchos resultados diferentes. Las prácticas son de alguna manera lo opuesto a los valores son concretos y realistas. Define los detalles de qué hacer las prácticas ayudan a los equipos a rendir cuentas ante los valores.

Por ejemplo la práctica de los espacios de trabajos informativos favorece la comunicación.

Transparente y sencilla y los principios son directrices específicas del dominio que cierran las brechas entre las prácticas y los valores.

Los valores de XP son comunicación simplicidad retroalimentación coraje y

respeto pero
echemos un vistazo a cada uno de ellos
con más detalles
iniciamos por la comunicación la falta
de comunicación impide que el
conocimiento fluya dentro de un equipo a
menudo cuando hay un problema alguien ya
sabe como resolverlo pero la falta de
comunicación les impide aprender sobre
el problema o contribuir a su solución
por lo tanto el problema termina
resolviéndose dos veces y en el proceso
generando muchos residuos
la simplicidad dice que siempre te
esfuerzas por hacer lo más simple que
funciona a menudo se malinterpreta y se
toma como la cosa más simple y punto
ignorando la parte de que funcione
también es crucial recordar que la
simplicidad es altamente contextual lo
que es simple para un equipo es complejo
para otro dependiendo completamente de
las habilidades la experiencia y el
conocimiento de cada equipo
en cuanto a la retroalimentación viene
muchas formas y tamaño cuando estás
programando en pareja los comentarios de
tu compañeros son comentarios vitales
También lo son las opiniones de otros
miembros del equipo sobre una idea
incluido el cliente que idealmente es
también miembro del equipo las pruebas
son otra fuente de retroalimentación
valiosa que más allá de los resultados
de las pruebas
aportan mucho valor al equipo de XP
si te está costando escribir pruebas es
probable que tu diseño sea demasiado
complejo escucha los comentarios y
simplifique su diseño algo que suena
como una gran idea podría no funcionar
también en la práctica por lo tanto el
código terminado también es una fuente
de retroalimentación al igual que un
producto implementado
si un equipo genera más comentarios de
los que puede manejar los comentarios
importantes pueden caer fuera del radar

en especial luego de reducir la velocidad y averiguar Qué causa el exceso de retroalimentación y arreglarlo en cuanto al coraje Ken Beck Define el coraje como una acción afectiva frente al miedo como ingeniero de software tiene mucho que temer y por lo tanto muchas oportunidades para mostrar coraje se necesita coraje para decir la verdad Especialmente los desagradables por ejemplo estimaciones honesta dar y recibir comentarios también requiere de mucho coraje y se necesita valor para evitar caer en la falacia del costo hundido y descartar una solución fallida que recibió inversiones sustanciales cuando hablamos del respeto una premisa fundamental de XP es que todo el mundo se preocupa por su trabajo ninguna cantidad de excelencia técnica puede salvar un proyecto si no hay cuidado y respeto cada persona es digna de dignidad valga la redundancia y respeto y Eso incluye por supuesto a las personas afectadas por un proyecto de desarrollo de software cuando usted y los miembros de su equipo se respetan y se preocupan el uno por el otro el cliente el proyecto y sus futuros usuarios todos ganan en cuanto a los principios proporcionan una guía más específica que los valores son pautas que iluminan los valores y los hacen más explícito y menos ambiguos por ejemplo basándose en el valor del coraje se podría concluir que es aconsejable abordar un gran cambio a la vez en el programa sin embargo el principio de baby steps nos dice que los grandes cambios son arriesgados por lo tanto quieres favorecer a los pequeños en su lugar en cuanto a la humanidad los humanos crean software para los humanos un hecho que a menudo se pasa por alto pero tener en cuenta las necesidades básicas fortalezas y debilidades humanas

crea productos que los humanos quieran en realidad usar y un entorno de trabajo que te da la oportunidad del logro y crecimiento y el sentimiento de pertenencia y la seguridad básica en su lugar donde tienes más fácilmente en cuenta las necesidades de los demás cuando nos referimos a la economía en XP los equipos tienen en cuenta las realidades económicas del desarrollo de software todo el tiempo evalúan constantemente los riesgos económicos y las necesidades del proyecto por ejemplo implementarían historias de usuarios de acuerdo con su valor comercial en lugar de sus preocupaciones técnicas y en cuanto al beneficio mutuo después de XP evitas soluciones que beneficien a una parte en detrimento de la otra por ejemplo las especificaciones extensas podrían ayudar a otra persona a entenderlo pero te aleja de implementarlo y retrasa toda la implementación para tus usuarios una solución mutuamente beneficiosa es utilizar pruebas de aceptación automatizadas usted recibe comentarios inmediatos sobre su implementación sus colegas obtienen especificaciones precisas en el código y los usuarios obtienen sus características lo antes posible además todos obtienen una red de seguridad contra las reingresiones hablemos de la auto similitud si una solución dada funciona a un nivel también podría funcionar a un nivel más alto o más bajo a nivel de desarrollador los programadores reciben comentario de su trabajo utilizando el enfoque test First a nivel de equipo la canalización de integración continua Integra construye y prueba el código varias veces al día y a nivel de organización los ciclos semanales y trimestrales permiten a los equipos obtener comentarios y mejorar su trabajo Cada vez que sea necesario Hablemos del principio de la mejora de

acuerdo con este principio los equipos no se esfuerzan por la perfección de una implementación inicial si no por una lo suficientemente buena y luego aprenderla y mejorarla continuamente con comentarios de usuarios reales si hablamos de la diversidad usted y sus compañeros de trabajo Se benefician de una diversidad de perspectivas habilidades y actitudes tal vez diversidad a menudo conduce a conflicto pero eso está bien el conflicto y el desacuerdo son oportunidades para que surjan mejoras ideas nuevas cuando todos están jugando el valor del coraje y del respeto en cuanto al coraje para expresar puntos de vistas Opuestos respeto a expresarlo de una manera civilizada y empática y todo esto es un ejercicio de comunicación efectiva si hablamos de la reflexión los grandes equipos reflexionan sobre su trabajo y analizan Cómo ser mejores XP ofrece muchas oportunidades para eso no sólo en sus ciclos semanales y trimestrales sino en todas las prácticas que promueve los sentimientos son importantes a tener en cuenta Además del Análisis Lógico tu instinto puede informarte antes de que puedas razonar sobre algo y también puedes hablar con personas no técnicas pueden hacer preguntas que habrán posibilidades completamente nuevas y si hablamos del flujo las metodologías tradicionales de desarrollo de software tienen fases discreta que duran mucho tiempo y tienen poca retroalimentación y oportunidades de corrección de rumbo en cambio el desarrollo de software XP se produce en actividades que ocurren todo el tiempo en un flujo de valor constante si hablamos de la oportunidad debemos saber que los problemas son inevitables en el desarrollo de software sin embargo cada problema es una oportunidad de mejora aprende a mirarlos de una manera totalmente distinta y es mucho más

probable que encuentres soluciones creativas y orientada a objetivos que también sirvan para evitar que vuelvan a suceder

cada uno de estos es una solución de una sola línea de defensa

juntos un muro prácticamente impenetrable

en cuanto al fracaso el fracaso no es un desperdicio cuando resulta en conocimiento actuar y aprender

rápidamente lo que no funciona es mucho más productivo que la inacción causada por la indecisión de elegir entre muchas opciones

respecto a la calidad a menudo la gente piensa que hay un dilema entre la calidad y la velocidad

es todo lo contrario impulsar mejoras de calidad es lo que te hace ir más rápido

por ejemplo la refactorización cambiar la estructura del código sin alterar su comportamiento es una práctica que hace que el código sea más fácil de entender y cambiar

como resultado es menos probable que introduzca defectos en el código lo que permite entregar más valor al no tener que corregir errores

en cuanto a los baby steps las grandes cambios son arriesgado XP mitiga ese riesgo haciendo cambios en pequeños pasos en todos los niveles los programadores escriben códigos en pequeños pasos utilizando el desarrollo basado en pruebas integran su código en la línea principal varias veces al día

el lugar de solo cada una pocas semanas o incluso meses el proyecto en sí se produce en ciclos cortos en lugar de una fase duradera o de Gran duración

y en cuanto a la responsabilidad aceptada en XP la responsabilidad debe ser aceptada nunca asignada la responsabilidad debe ir acompañado de la oportunidad para tomar decisiones sobre lo que usted es responsable también se aplica lo contrario no quiere que la

gente tome decisiones y no tienen que vivir en sus consecuencias

En definitiva la programación extrema siendo parte de la agilidad se trata de abrazar y dar la bienvenida al cambio en lugar de seguir planes rígidos se trata de Un diseño iterativo en lugar de un gran diseño por adelantado XP difiere drásticamente de las metodologías tradicionales es decir en cascada al evitar fases largas o procesos demasiados estructurados y para responder Una última pregunta que en qué se diferencia la programación extrema XP de las otras metodologías ágiles pues pudiéramos decir que la programación extrema por naturaleza tiene mucho en común con las otras metodologías ágiles Pero también es única Entre ella la mayoría de otras metodologías de desarrollo no dicen mucho si es que dicen algo sobre lo sobre cómo hacer el trabajo XP por otro lado es muy obstinado cuando se trata de eso y pone un gran énfasis en las prácticas de ingeniería de software más adelante estaremos planteando la diferencia de XP versus a cada una de las demás metodologías o por lo menos las más usadas como por ejemplo scrum kanban entre otras metodologías

[Música]

Pues bien mis estimados Muchísimas gracias por llegar hasta este punto del vídeo si el vídeo Te gustó como siempre voy a pedirte que me lo dejes saber con un buen like que te suscribas a este canal para que siga recibiendo este tipo de contenido aprieta la campanita para que te avise YouTube cuando subamos un próximo vídeo y dicho esto señores nos vemos en el próximo vídeo Bye

[Música]