4) b) En primer lugar se relacionará el ciclo de desarrollo TDD y los valores y prácticas de XP, se puede decir que el primero está directamente ligado a la simplicidad, ya que TDD consiste en tareas sencillas y cortas, generar pruebas unitarias que fallen para luego generar el código que haga que la prueba sea satisfactoria, una vez llegado a este punto se busca mejorar el código refactorizando el mismo y una vez realizado esto se corren nuevamente las pruebas, con esto se pueden encontrar fallas de forma más rápida y darle una posible solución ni bien se encuentra el inconveniente. Las pruebas generadas al principio del ciclo permiten identificar qué es lo que debe realizar el sistema, en cierta forma es una forma de comunicación entre los desarrolladores de la funcionalidad que se debe cubrir o se cubrió con los casos de test realizados. En la fase final de cada ciclo, ya se obtiene un resultado que puede ser evaluado por el cliente final, de esta forma puede aportar más información para el desarrollo de la funcionalidad o dar por satisfecho dicho punto. El valor respeto se puede asociar a la etapa de refactorización, en estos casos los desarrolladores deben estar abiertos a sugerencias en el cambio del código que desarrollaron.

En segundo lugar, se asociará el ciclo de desarrollo TDD con las propiedades de la metodología Crystal. TDD al estar basado en pruebas, las mismas podrían significar ser pequeñas funcionalidades a mostrar al cliente, esto se puede asociar a la propiedad de entregas frecuentes. La mejora reflexiva podría asociarse al proceso de refactorización ya que en este se podría dar diálogo entre los desarrolladores que permite aumentar la calidad del código desarrollado. La comunicación osmótica puede estar relacionada a la creación de casos de test y a la refactorización de código, ambos son formas de comunicar comportamiento y de aprender. La seguridad personal puede estar implícita en la refactorización de código, más en los casos de programación de a pares, en los cuales se debería poder corregir y aportar conocimientos al otro desarrollador, a su vez, puede aparecer en las entregas al cliente en las cuales debería haber una comunicación fluida y abierta. El enfoque debe aparecer en la definición de los casos de test que cumplan los criterios de aceptación propuestos por el cliente a un determinado requerimiento, el desarrollador tiene que conocer a la perfección la funcionalidad solicitada por el cliente para poder enfocarse correctamente en el armado del test y en su correspondiente código que haga que el test pase. La propiedad fácil acceso a usuarios expertos, está asociado a la correcta generación de test unitarios, el usuario experto debería facilitar el entendimiento de la funcionalidad para poder generar el test unitario correspondiente para pasar el código generado para dicha funcionalidad.

Por último se asociará Crystal con los valores y las prácticas XP. Se puede trazar un paralelismo entre las propiedades del primero y los valores y las prácticas del segundo. La propiedad entregas frecuentes de Crystal esta fuertemente ligada a la práctica versiones pequeñas, ambas tienen como fin entregar al cliente pequeñas funcionalidades funcionando en períodos acotados de tiempo. Mejora reflexiva está asociada como mencionamos anteriormente a la refactorización, asi como también a valores de XP como comunicación y retroalimentación o feedback, en todos los casos se busca a través de la comunicación mejorar el desarrollo del producto final, para que el mismo al final sea lo solicitado por el cliente. Comunicación osmótica, está asociada a la programación de a pares, con esto se da una comunicación implícita al momento de desarrollar, lo mismo sucede con la práctica denominada normas de codificación, ya que al establecer un estándar o patrón para codificar es una forma de comunicación que permite identificar que hace una función o variable de alguna parte del código. La propiedad enfoque de Crystal . En XP, el cliente al formar parte del equipo de desarrollo implica que la propiedad fácil acceso a usuarios expertos sea de alguna forma también parte de dicha metodología, ya que se puede comunicar constantemente con el cliente y obtener feedback para verificar que se estén haciendo el desarrollo de la forma correcta y que el cliente espera. La seguridad personal puede vincularse al valor respeto de XP, ya que ambos sugieren que las opiniones de las personas del equipo y el trabajo realizado por ellas sean valorados y respetados. Por último, la propiedad de Crystal que trata del entorno técnico con pruebas automatizadas, gestión de la configuración e integración continua está fuertemente asociada a las prácticas testing continuo e integración continua que establece XP, en estos casos se busca que cada incremento del programa entregado funcione y además se de acuerdo a lo solicitado por el cliente. Se puede pensar que antes de meterse de lleno en la programación extrema, se puede utilizar la metodología Crystal Clear, la cual tiene varios conceptos de XP sin llevarlos a utilizar al extremo como este caso sino que es más tolerante y no tan prescriptivo. Luego de incorporar ciertos conceptos de Crystal, y si el equipo no es de un tamaño superior a 12 personas, se podría en caso de querer apuntar a mejorar la calidad del desarrollo utilizar XP.

c) CMMI para Desarrollo, al igual que las metodologías ágiles, establece una serie de prácticas. Las mismas permiten cubrir varios procesos utilizados en el desarrollo y mantenimiento, en el caso de las ágiles nos indican las mejores formas para utilizar una metodología de desarrollo ágil. En CMMI existen dos tipos de representaciones, uno denominado continua y el otro por etapas, ambas representaciones buscan la mejora de sus procesos, si bien esto puede parecer en contra de uno de los principios de las metodologías ágiles, en donde se prioriza individuos e interacciones sobre procesos y herramientas, se puede decir que la finalidad en ambos casos es la misma aportar calidad al resultado final y ser más eficientes en el camino hacia el objetivo que se busca en cada caso.

CMMI y las metodologías ágiles son compatibles, mientras que el primero se centra en qué debería hacer el proyecto, el según aporta herramientas que indican cómo se debe realizar el proyecto.

En este apartado se propondrá como las metodologías ágiles **aportan soluciones a** las diferentes Áreas de Proceso que establece CMMI. La Gestión de Requerimientos (REQM), con respecto a la metodología SCRUM puede estar ligada al armado del Product Backlog, a la selección de historias de usuario para el próximo sprint. Se debe tener trazabilidad de los requerimientos enunciados, en el caso de XP, las prácticas Planificación y Metáforas son las que estarían asociadas a esta Área de Proceso, con las mismas se define lo que el cliente y se utilizan ciertas frases que permiten al cliente comprender rápidamente la funcionalidad a desarrollar.

La Planificación de Proyecto (PP), está ligada a las mismas prácticas especificadas en el párrafo anterior, se puede agregar que se debe hacer foco en la comunicación constante con el cliente, como enuncia XP, para establecer correctamente las prioridades de los requerimientos solicitados.

Con respecto a la Monitorización y Control del Proyecto (PMC), SCRUM por ejemplo, ofrece una ceremonia denominada Daily Meeting, en la cual se puede verificar el avance diario del proyecto y las complicaciones que pueden ir surgiendo lo cual va a permitir toman acciones correctivas para no desviarse de lo planificado.

Medición y Análisis (MA), se puede utilizar un artefacto que ofrece scrum denominado Burndown Chart que permite ver si se están cumpliendo las historias de usuario de acuerdo a lo esperado. Con esto se puede ver el avance del proyecto en cuanto a funcionalidades desarrolladas y tomar las acciones necesarias de acuerdo a posibles desvíos. Otra artefacto que ofrece scrum es la velocity de cada sprint planning, con esto se podría analizar cuántas user story podemos realizar en cada sprint de acuerdo a los stories point que tienen asignado. En XP, hay un rol encargado de realizar un seguimiento de las métricas en cuanto al avance del proyecto, defectos surgidos y otras variables, este se denomina tracker, y es quien propone mejoras al proyecto de acuerdo al análisis que realiza sobre las métricas.

3) Para el caso de QuaTeams, le convendría aplicar CMMI, con esto se asegurarían una mejora en la calidad de los procesos para el desarrollo de software que tendría como consecuencia la mejora en la calidad de los productos y servicios que desarrolle. De aplicar este modelo seguramente se hubiera reducido la cantidad de horas de re-trabajo realizado. Al ser una empresa pequeña, si se definen correctamente los procesos no se depende de las personas que en ella trabajan, por lo que si alguna se va de la empresa si se tienen los procesos bien definidos la estructura de la misma no se vendría abajo y podría sostenerse y seguir su curso.

Se puede suponer que QuaTeams cuenta con procesados inmaduros, según el SEI son considerados así ya que son procesos improvisados, procesos que tienen como fin cumplir con costos y fechas comprometidas y procesos cuya calidad no se puede predecir. Esto se puede ver en el caso de los problemas surgidos en la implementación del CRM en IIF, ante los problemas surgidos, sus procesos no tenían un plan de mitigación de riesgos como fue el caso del intermediario entre la interfaz de usuario y la base de datos denominado Castor. Las consultas a la base de datos tardaban alrededor de unos 6 a 10 minutos, este hecho no fue contemplado por lo que no tenían una alternativa rápida para solucionar el inconveniente y pudo haberse producido un estado caótico ante la falta de respuesta a una nueva alternativa viable. Por otro lado, el costo del producto que ofrecían terminó siendo bastante ajustado de acuerdo a las horas estimadas de la implementación.

Igualmente se tiene que saber que CMMI permite mejorar lo que ya se tiene, no es un manual que dice qué se debe hacer, sino que establece ciertas buenas prácticas que las organizaciones pueden incluir en sus procesos para mejorar los mismos.

Se recomendarìa utilizar la representación continua

(RD)

(REQM)

(TS)

(VAL)

(VER)

(CM)

(RSKM)

**En Zara, se aconsejaría acreditar en CMMI en el área de TI con un nivel de madurez 2: Gestionado. Esto con el fin de hacer foco en los procesos ya que carecen de los mismos y podría evitar complicaciones a futuro, a partir de esto se debe generar documentación apropiada que permita avanzar en el desarrollo de software y también se deben realizar estimaciones adecuadas que denoten el trabajo que debe realizar el área para la evolución de las aplicaciones. En el caso de Zara, el departamento de TI parece una estructura separa del resto de la empresa, formalizar sus procesos podría significar incorporar en cierta forma el departamento en esta estructura de empresa y a su vez destinar un presupuesto acorde al área.**

**EL diagnostico realizado, es que no se evidencia conocimiento claro de los costos que se realizan en esta área, esto se puede deber principalmente a la mala planificación de los proyectos y a la falta de integración de este departamento con el resto de la compañía. Al suponer una fuerte dependencia de Salgado y Castellano, se puede decir que el desarrollo de software está atado a determinadas personas y no hay un proceso que sume al proceso de desarrollo de software. Por otro lado, se puede mejorar el proceso de comunicación con proveedores, para conocer constantemente las novedades de las actualizaciones tecnológicas que estos implementen para así verificar que las aplicaciones de Zara sigan siendo compatibles con la tecnología ofrecida.**

**En este caso las cuatro áreas de proceso a trabajar serían (SAM) y (PP) y (REQM) y (IPM).**

**Primero se realizaría un trabajo en el armado de planificación de los proyectos de software y en la involucrción de las demás áreas en el trabajo del departamento de TI. Con esto se buscaría integrar el departamento de TI con el resto de la empresa para que durante el desarrollo del software se pueda obtener feedback constante de las áreas involucradas en los cambios a realizar, a su vez se busca planificar proyectos de desarrollo que puedan permitir establecer el presupuesto a utilizar en dicho desarrollo, también se debe definir el alcance e informarlo a las áreas afectadas por el mismo, para esto se necesita un fuerte conocimiento de las necesidades de las áreas que utilizan el software.**

**4) a) En primer lugar se establecerá una relación entre las fases y flujo de UP y los cinco pasos enunciados por Royce, estos son: el diseño del programa viene primero; documentar el diseño; hacerlo dos veces; planificar, controlar y monitorear el testeo e involucrar al cliente. Se pueden incluir estos pasos en las diferentes fases de UP reemplazando a los flujos que esta metodología enuncia por los de Winston Royce. Esto se puede hacer ya que las tareas que implican los pasos como los flujos de trabajo de UP en términos generales son las mismas.**

**El primer paso, que menciona Royce, puede tener mucha carga de trabajo dentro de la primera etapa UP denominada “Inicio”, aquí se define la arquitectura de la aplicación, a nivel general sin entrar en detalles. Luego, en la fase “Elaboración” se puede tener la mayor cantidad de esfuerzo invertido, ya que se debe diseñar la arquitectura correspondiente al software a desarrollar. En esta fase, también entraría en juego el segundo paso enunciado por Royce, se deben diseñar en primer lugar los casos de uso más críticos del sistema a desarrollar y se puede comenzar con el desarrollo de los mismos. Luego en esta misma etapa o fase se puede continuar con el tercer paso; refinando los casos de uso y agregando otros no tan críticos, como los realizados anteriormente, y comenzar con su respectivo desarrollo. Este tercer paso, también se realiza en gran medida en la fase de construcción.**

**El cuarto paso tendría su mayor carga en las últimas iteraciones de la etapa de construcción y en las primeras de la fase transición. Por último con respecto al involucramiento del cliente, se mantendría de forma constante en las primeras etapas y tendría mayores picos durante las últimas iteraciones de la fase construcción y las primeras de la fase transición.**

**En segundo lugar, se asociará las dimensiones angulares y radiales del modelo en espiral de Boehm con las fases y flujos de UP. Se puede pensar, que cada fase del proceso unificado contiene un modelo en espiral, en la primer fase los objetivos se pueden relacionar a definición de requerimientos, mientras más pequeño sea el espiral mayor debe ser la criticidad de los requerimientos relevados y definidos. Aquí también debe hacerse hincapié en los riesgos que puedan producirse en el proyecto. Luego a medida que se avanza en las iteraciones del ciclo en espiral se debe realizar un mayor refinamiento de los requerimientos planteados en esta fase. A medida que aumenta el ángulo del espiral es mayor la documentación generada en esta fase para la definición de los requisitos. Al finalizar cada vuelta se debe establecer con el cliente si las definiciones realizadas son correctas y relevar con el mismo los próximos requerimientos que se van a documentar.**

**El espiral de la fase elaboración, tendrá como primeros objetivos diseñar la arquitectura de la aplicación, y a medida que el espiral crezca se van definiendo los casos de uso de los más críticos a los menos relevantes, al final de cada iteración del espiral se verifican los mismos con el cliente para que sean aprobados y continuar con los siguientes, se definen los riesgos asociados a cada caso de uso y se deben ir mitigando los relacionados a los más críticos.**

**En la fase construcción se cuenta con un espiral en donde se empiezan a poner objetivos asociados al desarrollo del software, empezando a desarrollar los casos de uso más prioritarios y realizando en cada iteración las pruebas que cubran el desarrollo realizado en la misma.**

**En la última fase, los objetivos del espiral van a estar relacionados en primer medida a la capacitación de los usuarios y luego al control y seguimiento de defectos que surgen. Aquí los riesgos planteados no deberían ser críticos, ya que los mismos se debían mitigar en fases anteriores.**

4) a) En primer lugar se establecerá una relación entre las fases y flujo de UP y los cinco pasos enunciados por Royce, estos son: el diseño del programa viene primero; documentar el diseño; hacerlo dos veces; planificar, controlar y monitorear el testeo  e involucrar al cliente. Se pueden incluir estos pasos en las diferentes fases de UP reemplazando a los flujos que esta metodología enuncia por los de Winston Royce. Esto se puede hacer ya que las tareas que implican los pasos como los flujos de trabajo de UP en términos generales son las mismas.

El primer paso, que menciona Royce, puede tener mucha carga de trabajo dentro de la primera etapa UP denominada “Inicio”, aquí se define la arquitectura de la aplicación, a nivel general sin entrar en detalles. Luego, en la fase “Elaboración” se puede tener la mayor cantidad de esfuerzo invertido, ya que se debe diseñar la arquitectura correspondiente al software a desarrollar. En esta fase, también entraría en juego el segundo paso enunciado por Royce, se deben diseñar en primer lugar los casos de uso más críticos del sistema a desarrollar y se puede comenzar con el desarrollo de los mismos. Luego en esta misma etapa o fase se puede continuar con el tercer paso; refinando los casos de uso y agregando otros no tan críticos, como los realizados anteriormente, y comenzar con su respectivo desarrollo. Este tercer paso, también se realiza en gran medida en la fase de construcción.

El cuarto paso tendría su mayor carga en las últimas iteraciones de la etapa de construcción y en las primeras de la fase transición. Por último con respecto al involucramiento del cliente, se mantendría de forma constante en las primeras etapas y tendría mayores picos durante las últimas iteraciones de la fase construcción y las primeras de la fase transición.

En segundo lugar, se asociará las dimensiones angulares y radiales del modelo en espiral de Boehm con las fases y flujos de UP. Se puede pensar, que cada fase del proceso unificado contiene un modelo en espiral, en la primer fase los objetivos se pueden relacionar a definición de requerimientos, mientras más pequeño sea el espiral mayor debe ser la criticidad de los requerimientos relevados y definidos. Aquí también debe  hacerse hincapié en los riesgos que puedan producirse en el proyecto. Luego a medida que se avanza en las iteraciones del ciclo en espiral se debe realizar un mayor refinamiento de los requerimientos planteados en esta fase. A medida que aumenta el ángulo del espiral es mayor la documentación generada en esta fase para la definición de los requisitos. Al finalizar cada vuelta se debe establecer con el cliente si las definiciones realizadas son correctas y relevar con el mismo los próximos requerimientos que se van a documentar.

El espiral de la fase elaboración, tendrá como primeros objetivos diseñar la arquitectura de la aplicación, y a medida que el espiral crezca se van definiendo los casos de uso de los más críticos a los menos relevantes, al final de cada iteración del espiral se verifican los mismos con el cliente para que sean aprobados y continuar con los siguientes, se definen los riesgos asociados a cada caso de uso y se deben ir mitigando los relacionados a los más críticos.

En la fase construcción se cuenta con un espiral en donde se empiezan a poner objetivos asociados al desarrollo del software, empezando a desarrollar los casos de uso más prioritarios y realizando en cada iteración las pruebas que cubran el desarrollo realizado en la misma.

En la última fase, los objetivos del espiral van a estar relacionados en primer medida a la capacitación de los usuarios y luego al control y seguimiento de defectos que surgen. Aquí los riesgos planteados no deberían ser críticos, ya que los mismos se debían mitigar en fases anteriores.

**En último lugar, se relacionará el modelo en espiral con los pasos enunciados por Winston Royce en su artículo. En el tercer paso planteado por este último, se podría aplicar un modelo en espiral, donde el interior del mismo corresponde a un enfoque general de las tareas de análisis, diseño, construcción y testing. Con esto, al finalizar la primera iteración, se puede mostrar la versión inicial de la aplicación y a partir de la misma se puede ir refinando en las siguientes iteraciones cada etapa involucrada. Es decir, cada próxima iteración va a abarcar cada una de las etapas mencionadas en su totalidad, y de esta manera se van a poder definir riesgos más específicos y refinar cada etapa de modo de cumplir con los requisitos solicitados.**

4) b) En primer lugar se relacionará el ciclo de desarrollo TDD y los valores y prácticas de XP. Se puede decir que el primero está directamente ligado a la simplicidad, ya que TDD consiste en tareas sencillas y cortas, generar pruebas unitarias que fallen para luego generar el código que haga que la prueba sea satisfactoria, una vez llegado a este punto se busca mejorar el código refactorizando el mismo y una vez realizado esto se corren nuevamente las pruebas, con esto se pueden encontrar fallas de forma más rápida y darle una posible solución ni bien se encuentra el inconveniente.

Las pruebas, generadas al principio del ciclo, permiten identificar qué es lo que debe realizar el sistema. Se puede decir que es una forma de comunicación entre los desarrolladores de la funcionalidad que se debe cubrir o se cubrió con los casos de test realizados. En la fase final de cada ciclo, ya se obtiene un resultado que puede ser evaluado por el cliente final, de esta forma puede aportar más información para el desarrollo de la funcionalidad o dar por satisfecho dicho punto. El valor respeto se puede asociar a la etapa de refactorización, en estos casos los desarrolladores deben estar abiertos a sugerencias en el cambio del código que desarrollaron.

En segundo lugar, se asociará el ciclo de desarrollo TDD con las propiedades de la metodología Crystal Clear. Las pruebas, en las que se basa TDD, podrían significar ser pequeñas funcionalidades a mostrar al cliente, esto se puede asociar a la propiedad de entregas frecuentes que enuncia Crystal. La mejora reflexiva podría asociarse al proceso de refactorización, debido a que esta tarea puede significar un diálogo entre los desarrolladores, que permite aumentar la calidad del código desarrollado y adquirir conocimientos para futuros desarrollos. La comunicación osmótica puede estar relacionada a la creación de casos de test y a la refactorización de código, ambos son formas de comunicar comportamiento y de aprender, sin necesidad de establecer un diálogo entre los programadores. La seguridad personal puede estar implícita en la refactorización de código, más en los casos de programación de a pares, en los cuales se debería poder corregir y aportar conocimientos al otro desarrollador, a su vez, puede aparecer en las entregas al cliente en las cuales debería haber una comunicación fluida y abierta. El enfoque debe aparecer en la definición de los casos de test que cumplan los criterios de aceptación propuestos por el cliente a un determinado requerimiento, el desarrollador tiene que conocer a la perfección la funcionalidad solicitada por el cliente para poder enfocarse correctamente en el armado del test y en su correspondiente código que haga que el test pase. La propiedad fácil acceso a usuarios expertos, está asociado a la correcta generación de test unitarios, el usuario experto debería facilitar el entendimiento de la funcionalidad para poder generar el test unitario correspondiente para pasar el código generado para dicha funcionalidad. **La propiedad entorno técnico con pruebas automatizadas, gestión de la configuración e integración continua, está totalmente ligada a una extensión del modelo de ciclo de vida TDD denominado ATDD; ya que este último está orientado a pruebas funcionales, lo cual implica que siempre haya funcionalidad tangible para el cliente, esto lo permite la integración continua, y además para utilizar esta metodología se requiere automatizar las pruebas de aceptación de usuario indicadas en las historias de usuario.**

Por último se asociará Crystal Clear con los valores y las prácticas  XP. Se puede trazar un paralelismo entre las propiedades del primero y los valores y las prácticas del segundo. La propiedad entregas frecuentes de Crystal está fuertemente ligada a la práctica versiones pequeñas, ambas tienen como fin entregar al cliente funcionalidades en períodos acotados de tiempo. La mejora reflexiva está asociada, como mencionamos anteriormente, a la refactorización, así como también a valores de XP como comunicación y retroalimentación o feedback, en todos los casos se busca a través de la comunicación mejorar el desarrollo del producto, para que el mismo al final sea lo solicitado por el cliente. Comunicación osmótica está asociada a la programación de a pares, con esto se da una comunicación implícita al momento de desarrollar, lo mismo sucede con la práctica denominada normas de codificación, ya que al establecer un estándar o patrón para codificar es una forma de comunicación que permite identificar qué hace una función o variable de alguna parte del código. La propiedad enfoque de Crystal**, se relaciona al valor comunicación de la programación extrema, esto se debe a que cada integrante del equipo de desarrollo tiene que conocer claramente las tareas que debe realizar, y el diálogo es la forma de hacer efectivo esto**. En XP, el cliente al formar parte del equipo de desarrollo implica que la propiedad fácil acceso a usuarios expertos, sea también parte de dicha metodología, ya que se puede comunicar constantemente con este y obtener feedback para verificar que se esté haciendo el desarrollo solicitado. La seguridad personal puede vincularse al valor respeto de XP, ya que ambos sugieren que las opiniones de las personas del equipo y el trabajo realizado por ellas sean valorados y respetados. Por último, la propiedad de Crystal que trata del entorno técnico con pruebas automatizadas, gestión de la configuración e integración continua, está asociada a las prácticas testing continuo e integración continua que establece XP, en estos casos se busca que cada incremento del programa entregado funcione y además cumpla con lo solicitado por el cliente.

Se puede pensar que antes de meterse de lleno en la programación extrema, se puede utilizar la metodología Crystal Clear, la cual tiene varios conceptos de XP sin llevarlos a utilizar al extremo, sino que es más tolerante y no tan prescriptivo. Luego de incorporar ciertos conceptos de Crystal, y si el equipo de desarrollo no es de un tamaño superior a 12 personas, en caso de querer implementar prácticas que puedan mejorar la calidad del desarrollo, se podría pasar a utilizar XP.

c) CMMI para desarrollo, al igual que las metodologías ágiles, establece una serie de prácticas, las mismas permiten cubrir varios procesos utilizados en los proyectos. En el caso de las ágiles, nos indican las mejores formas para utilizar una metodología de desarrollo ágil.

En CMMI existen dos tipos de representaciones, una denominada continua y la otra por etapas, ambas representaciones buscan la mejora de sus procesos. Si bien esto puede parecer en contra de uno de los principios que enuncia el manifiesto ágil, en donde se prioriza individuos e interacciones por sobre procesos y herramientas, se puede decir que la finalidad en ambos casos es la misma: aportar calidad al resultado final y ser más eficientes en el camino hacia el objetivo que se busca en cada caso. CMMI y las metodologías ágiles son compatibles, mientras que el primero se centra en qué debería hacer el proyecto, el segundo aporta herramientas que indican cómo se debe realizar el proyecto.

En este apartado se propondrá cómo las metodologías ágiles pueden aportar soluciones a las diferentes Áreas de Proceso (KPAs) que establece CMMI. Por ejemplo, la metodología Scrum puede aportarle diferentes herramientas a la mejora del área de proceso Gestión de Requerimientos (REQM), estas pueden ser el armado del Product Backlog y la selección y priorización de historias de usuario para el próximo sprint. Continuando con ésta área de proceso, se debe tener trazabilidad de los requerimientos enunciados, en el caso de XP, las prácticas Planificación y Metáforas son las que le aportan herramientas, con las mismas se define lo que el cliente desea y se utilizan ciertas frases que permiten a este comprender rápidamente la funcionalidad a desarrollar.

La Planificación de Proyecto (PP), puede estar ligada a las mismas prácticas especificadas en el párrafo anterior. Se puede agregar que se debe hacer foco en la comunicación constante con el cliente, como enuncia XP, para establecer correctamente las prioridades de los requerimientos solicitados.

Con respecto a la Monitorización y Control del Proyecto (PMC), SCRUM por ejemplo, ofrece una ceremonia denominada Daily Meeting, en la cual se puede verificar el avance diario del proyecto y las complicaciones que pueden ir surgiendo lo cual va a permitir toman acciones correctivas para no desviarse de lo planificado.

Medición y Análisis (MA), se puede utilizar un artefacto que ofrece scrum denominado Burndown Chart que permite ver si se están cumpliendo las historias de usuario de acuerdo a lo esperado. Con esto se puede ver el avance del proyecto en cuanto a funcionalidades desarrolladas y tomar las acciones necesarias de acuerdo a posibles desvíos. Otra artefacto que ofrece scrum es la velocity de cada sprint planning, con esto se podría analizar cuántas user story podemos realizar en cada sprint de acuerdo a los stories point que tienen asignado. En XP, hay un rol encargado de realizar un seguimiento de las métricas en cuanto al avance del proyecto, defectos surgidos y otras variables, este se denomina tracker, y es quien propone mejoras al proyecto de acuerdo al análisis que realiza sobre las métricas.

3) Para el caso de QuaTeams, le convendría aplicar CMMI, con esto se asegurarían una mejora en la calidad de los procesos para el desarrollo de software que tendría como consecuencia la mejora en la calidad de los productos y servicios que desarrolle. De aplicar este modelo seguramente se hubiera reducido la cantidad de horas de re-trabajo realizado. Al ser una empresa pequeña, si se definen correctamente los procesos no se depende de las personas que en ella trabajan, por lo que si alguna se va de la empresa si se tienen los procesos bien definidos la estructura de la misma no se vendría abajo y podría sostenerse y seguir su curso.

Se puede suponer que QuaTeams cuenta con procesados inmaduros, según el SEI son considerados así ya que son procesos improvisados, procesos que tienen como fin cumplir con costos y fechas comprometidas y procesos cuya calidad no se puede predecir. Esto se puede ver en el caso de los problemas surgidos en la implementación del CRM en IIF, ante los problemas surgidos, sus procesos no tenían un plan de mitigación de riesgos como fue el caso del intermediario entre la interfaz de usuario y la base de datos denominado Castor. Las consultas a la base de datos tardaban alrededor de unos 6 a 10 minutos, este hecho no fue contemplado por lo que no tenían una alternativa rápida para solucionar el inconveniente y pudo haberse producido un estado caótico ante la falta de respuesta a una nueva alternativa viable. Por otro lado, el costo del producto que ofrecían terminó siendo bastante ajustado de acuerdo a las horas estimadas de la implementación.

Igualmente se tiene que saber que CMMI permite mejorar lo que ya se tiene, no es un manual que dice qué se debe hacer, sino que establece ciertas buenas prácticas que las organizaciones pueden incluir en sus procesos para mejorar los mismos.

En Zara, se aconsejaría acreditar en CMMI en el área de TI. El diagnóstico realizado, es que no se evidencia conocimiento claro de los costos que tiene esta área, esto se puede deber principalmente a la mala planificación de los proyectos y a la falta de integración de este departamento con el resto de la compañía. Al suponer una fuerte dependencia de Salgado y Castellano, se puede decir que el desarrollo de software está atado a determinadas personas y no hay un proceso que sume calidad al desarrollo de software. Se puede suponer que no se cuenta con una herramienta que maneje las diferentes versiones de software implementadas, por lo que se debería tener registro de cada uno de los releases con su correspondiente versión. Por otro lado, se puede mejorar el proceso de comunicación con proveedores, para conocer constantemente las novedades de las actualizaciones tecnológicas que estos implementen y así verificar que las aplicaciones de Zara sigan siendo compatibles con la tecnología ofrecida.

Se propondría utilizar una representación continua. Antes de establecer en qué áreas de proceso se debe abordar, se realizará una descripción de qué se debe realizar. Primero se realizaría un trabajo en el armado de planificación de los proyectos de software, y en la involucración de las demás áreas en el trabajo del departamento de TI. Con esto se buscaría integrar el departamento de TI con el resto de la empresa, para que durante el desarrollo del software se pueda obtener feedback constante de las áreas involucradas en los cambios a realizar. A su vez se busca planificar proyectos de desarrollo que puedan permitir establecer el presupuesto a utilizar en dicho desarrollo, y se debe definir el alcance e informar a las áreas afectadas por el mismo; para esto se necesita un fuerte conocimiento de las necesidades de las áreas que utilizan el software. Para tener un control de los cambios realizados, se debe contar con un versionador que permita tener un histórico de los cambios efectuados. Por último se podría mejorar en el proceso de comunicación con los proveedores, para siempre contar con las novedades de las actualizaciones que estos realizan en sus productos.

En este caso las cinco áreas de proceso a trabajar serían (CM)(SAM) y (PP) y (REQM) y (IPM).

**El dinamismo de la empresa puede ser un obstáculo para llevar a cabo esta implementación, no se tiene el tiempo necesario para enfocarse en procesos que puedan realentizar sus procesos veloces de producción y puesta en venta. Los facilitadores que se puede encontrar, es que la empresa puede ver en esta implementación una mejora a la hora de luego adaptarse a los cambios que establece el mercado, con procesos definidos en TI y mayor conocimiento de quienes allí trabajen el sistema puede adaptarse más fácilmente al dinamismo que ofrece la industria textil.**