



Facultad de Ciencias
UNAM

Metodología de desarrollo de software: Crystal Clear

Oscar Eduardo Villa Chio
Jorge Luis García Flores
Enrique Antonio Bernal Cedillo
Víctor Zamora Gutiérrez

Riesgo Tecnológico
6 Semestre
22 de abril del 2016

1 Introducción

1.1 Objetivo general

El objetivo general de este trabajo es introducir la metodología Crystal Clear y explicar sus usos a través de varios ejemplos.

1.2 Objetivos específicos

Mostrarle a los alumnos de la clase de riesgo tecnológico cómo trabajar con la metodología Crystal Clear.

Otorgar una herramienta para terminar trabajos de pequeño alcance en tiempos razonables.

1.3 Alcance del trabajo

Clase de Riesgo Tecnológico.

2 Creador

Alistair Cockburn, es un computólogo de ciencias de la computación, fue uno de los firmantes del manifiesto de metodologías ágiles como también creador de su propia familia de metodologías llamada Cristal Clear a mediados de los años 90's las cuales han sido un gran aporte para la ingeniería de software. En su libro, Alistair explica que dicha metodología fue desarrollada para la elaboración de proyectos pequeños (de un aproximado de 8 integrantes como máximo, dentro de un área en común). Para obtener esta metodología, Alistair realizó investigación sobre grupos pequeños de desarrollo que cumplieran con éxito lo requerido.

Tomando en cuenta esto, entonces los puntos principales que las metodologías Crystal toman en cuenta son los siguientes:

El aspecto humano del equipo

El tamaño del equipo (o de componentes)

Comunicación directa entre componentes o integrantes

Políticas a seguir distintas

Contar con un único espacio físico de trabajo, para la tarea específica a realizar.

2.1 Manifiesto ágil

Entre febrero 11-13 del año 2001 se llevó a cabo una reunión organizada por Kent Beck en la cual reunió a 17 especialistas en modelos de desarrollo de software, en esta reunión se creó el término de Metodologías ágiles como alternativa a las metodologías formales que existían en la época como SCRUM y DSDM. Los involucrados en esta reunión fueron conocidos como firmantes del manifiesto ágil, los cuales fueron:

- Kent Beck
- Alistair Cockburn
- Mike Beedle
- Arie van Bennekum
- Ward Cunningham

- James Grenning
- Martin Fowler
- Jim Highsmith
- Andrew Hunt
- Ron Jeffries
- Jon Kern
- Brian Marick
- Robert Cecil Martin
- Steve Mellor
- Ken Schwaber
- Jeff Sutherland
- Dave Thomas

Metodologías ágiles

Dicha familia forma parte de la categoría de Metodologías ágiles de desarrollo de software, las cuales tienen como principales características:

Ser una alternativa a las metodologías tradicionales de desarrollo de software.

(Debido a su antigüedad.)

Valorar a las personas e interacciones por encima de las herramientas, procesos y productos.

Que el software aparte de funcionar cuente con una documentación comprensiva

Reducir la cantidad de intermediarios en el proyecto, y mantener la comunicación con el cliente durante el desarrollo

Reducir el tiempo utilizado en la toma de decisiones, para evitar pérdidas importantes de tiempo.

Respuesta ágil ante cualquier cambio ya sea en requerimientos, herramientas de desarrollo, el equipo y demás cuestiones.

[Tomado de:
<https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>]

Desarrollo de la metodología Crystal Clear

Fue de esta manera que se encontró que varios equipos realizaban las mismas acciones:

Sentar a la gente cerca. Que se comuniquen seguido y en un ambiente de amabilidad y de buena gana.

Alejar los temas burocráticos lo más posible de los desarrolladores.

Dejar que los desarrolladores diseñen el sistema y tomen las decisiones que consideren apropiadas.

Tener a un usuario involucrado en el proyecto.
(Este ayuda a encontrar errores, realizar pruebas y perfeccionar el diseño del sistema)

Tener un conjunto de pruebas de regresión automatizadas para asegurarse de que el software funciona correctamente en cada paso del desarrollo.

Producir funcionalidad entregable desde temprano y seguido (?).

Con esto hizo que las metodologías Crystal tuvieran un enfoque importante en:

Gente

Interacción

Comunidad

Habilidades

Talentos

Comunicación

Metodologías Crystal

La metodología Crystal Clear se puede describir como sigue:

El diseñador principal junto con otras 2 a 7 personas trabajan en un espacio pequeño como una habitación (o varias habitaciones adyacentes) con muestras visuales de información (como pizarrones) sin ninguna distracción y con acceso directo a usuarios clave.

Deben entregar código usable y probado aproximadamente cada 3 meses o menos a los clientes dependiendo de la duración del proyecto.

Por lo que la integración de componentes del proyecto también debe realizarse continuamente.

(Esto para mantener la comunicación tanto con el equipo como con el cliente, la cual es esencial para no perder tiempo)

Gracias a estas entregas frecuentes de prototipos, es posible mejorar por medio de retroalimentación con el cliente, esta es otra de las características importantes de la metodología Crystal Clear.

La metodología Crystal Clear no pretende ser la mejor, sino tan sólo ser "suficiente" (No asegura un producto perfecto, pero sí funcional y sin sacrificar el tiempo de los involucrados).

2.2 Familia Crystal

Alistair desarrolló diferentes metodologías basándose en los equipos de trabajo que requerían diferentes estrategias para resolver diferentes problemas. Para esto Alistair usó una distinción de colores para cada miembro de la familia crystal para indicar el "peso", estos métodos son:

Crystal Clear (1-6)

Crystal Yellow (7-20)

Crystal Orange (21-40)

Crystal Orange Web (21-40)

Crystal Red (41-80)

Crystal Maroon (81-200)

Crystal Diamond (201-500)

Crystal Sapphire (800-superior)

En donde se indica el número de integrantes que debe tener el equipo de trabajo basandose en la metodología a usar.

La familia Crystal define roles para así mejorar el rendimiento en el desarrollo del software los cuales son:

Patrocinador:

Es el encargado en asignar los fondos necesarios par el proyecto, adems de mantener una visión a largo plazo para mantener las prioridades cómo también tener la facultada de tomar la decicisión de detener el proceso desarrollo.

Usuario Experto:

Esta encargado de realizar y especificar una lista en el cuál indica los objetivos a realizar, también del personal requerido para lograrlo y el como definir los casos de uso y los requerimientos del sistema.

Diseador:

Es la persona encargada en hacer los diseos de los sistemas principales, adems de mantener al equipo de trabaja en el mismo canal de comunicacin para hacer un desarrollo homogéneo, ya que el diseador cuenta con la experiencia necesaria para realizar estas tareas.

Diseador-Programador:

Es el encargado (junto con el diseador) de producir las plantillas o borradores de las pantallas a usar el software.

Experto en negocios:

Es el encargado en verificar que estrategias a usar o que políticas serán fijas en el desarrollo del software.

Coordinador:

Es el encargado de dar la estructura y del estado del proyecto a los patrocinadores adems de que facilita las conversaciones para reducir los conflictos.

Verificador:

Realiza las pruebas necesarias para verificar la funcionaidad del sistema, adems de estar cómo encargado de la realización de reportes.

Escritor:

Realiza los manuales de usuario, esta persona debe tener un vocabulario amplio y una gran fluides para la expresión de ideas para sintetizar y especificar claramente el cómo se debe usar el software dado.

(Identificar los roles y agregar mas informacion de estas paginas, que explican chido:

<http://es.slideshare.net/fvalerolujano/metodologias-giles-crystal-clear>

https://en.wikiversity.org/wiki/Crystal_Methods

http://ingenieriadesoftware.mex.tl/59189_Metodologia-Crystal.html

)

(Un "ejemplo concreto", o ejemplos (como dice la introduccion), o cambiar la introduccion)

(Agregar muchas fuentes mas confiables y asi :V como quiere waifu, bien referenciadas)

(Obviamente ampliar y mejorar la conclusion)

3 Conclusiones

Es una estrategia fiable para el desarrollo de productos de software ya que es fácil de entender, es efectiva para un ambiente de trabajo tranquilo y eficaz ya que promueve la comunicación cara a cara con cada miembro relacionado con el proyecto adems de poseer una estructura jerarquica bien definida para cada role de trabajo que competen a una interacción constante entre sí para evitar impactos tanto en calidad cómo de tiempo.

4 Bibliografia

5 Anexos

(Solo dejar en caso de requerirlos)