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 las metodologías *Crystal* y en particular la metodología *Crystal Clear*.

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 Contenido

Creador

Alistair Cockburn se dedica a las ciencias de la computación. Fue uno de los firmantes del *manifiesto de metodologías ágiles*, un documento muy importante para la ingeniería de software, que habla acerca de métodos ligeros de desarrollo. También fue creador de su propia familia de metodologías llamada *Crystal* a mitades de los años 90's, las cuales han sido un gran aporte para la ingeniería de software.
En particular, hablaremos de la tecnología *Crystal Clear*. En su libro, Alistair explica que dicha metodología fue desarrollada para la elaboración de proyectos pequeños (de 8 integrantes como máximo). Para obtener esta metodología, Alistair realizó investigación sobre grupos de desarrollo pequeños e identificó patrones en los grupos exitosos.
Los puntos principales que las metodologías *Crystal* toman en cuenta son los siguientes:

- El aspecto humano del equipo
- El tamaño del equipo o los componentes
- Comunicación directa entre componentes o integrantes
- Las políticas a seguir
- Que se cuente con un único espacio físico de trabajo para la tarea específica a realizar.

Manifiesto ágil

Del 11 al 13 de febrero del 2001 se llevó acabo una reunión organizada por Kent Beck. Se reunieron 17 especialistas en modelos de desarrollo de software, quienes crearon el término *Metodologías ágiles* como alternativa a las metodologías formales que existian en la época (por ejemplo *SCRUM* y *DSDM*). Los involucrados en esta reunión fueron conocidos como firmantes del manifiesto ágil. Ellos 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

La familia *Crystal* 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 desarollo de software.
- Valorar a las personas e interacciones por encima de las herramientas, procesos y productos.
- Que el software, además de funcionar, cuente con una documentación comprensiva.
- Tener un número bajo de intermediarios en el proyecto, y mantener la comunicación con el cliente durante el desarrollo.

- Utilizar poco tiempo en la toma de decisiones, para evitar pérdidas importantes de tiempo.
- Tener una respuesta ágil ante cualquier cambio, ya sea en requerimientos, herramientas de desarollo, el equipo u otras cuestiones.

Metodologías Crystal

Alistair desarrolló diferentes metodologías basandose 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". Estas metodologías son:

- Crystal Clear (1-6)
- Crystal Yellow (7-20)
- Crystal Orange (21-40)
- Crystal Orange Web (21-40)
- Crytal 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.

Las metodologías *Crystal* tienen un enfoque importante en:

- Gente
- Interacción
- Comunidad
- Habilidades
- Talentos
- Comunicación

Roles en la familia *Crystal*

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

- Patrocinador:
  Es el encargado de asignar los fondos necesarios para el proyecto, además de mantener una visión a largo plazo respecto a este. Tiene la capacidad de tomar la decicisión de detener el proceso de desarrollo.

- Usuario Experto:
  Es el encargado de hacer una lista en la cuál se indican los objetivos a realizar, el personal requerido para lograrlo, los casos de uso y los requerimientos del sistema.

- Diseñador:
  Es la persona encargada de hacer los diseños de los sistemas principales, además de mantener al equipo de trabajo en el mismo canal de comunicación para realizar un desarrollo homogéneo.

- Diseñador-Programador:
  Es el encargado (junto con el diseñador) de producir las plantillas o borradores de las pantallas del software.

- Experto en negocios:
  Es el encargado de analizar estrategias a usar. Es quien define qué políticas serán fijas en el desarollo del software.

- Coordinador:
  Es el encargado de mostrar la estructura y el estado del proyecto a los patrocinadores . Se encarga de reducir los conflictos.

- Verificador:
  Realiza las pruebas necesarias para verificar la funcionalidad del sistema y se encarga de elaborar reportes.

- Escritor:
  Realiza los manuales de usuario. Esta persona debe tener un vocabulario amplio y una gran fluidez para la expresión de ideas. Debe ser capaz de especificar claramente el cómo se debe usar el software dado.

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. Esto 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 de manera frecuente.

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 (puede ser menos dependiendo de la duración del proyecto). 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 libre de los involucrados.

3 Conclusiones

Es una estrategia fiable para el desarrollo de productos de software ya que es fácil de enteder y es efectiva para un ambiente de trabajo tranquilo y eficaz. Promueve la comunicación cara a cara con cada miembro relacionado con el proyecto además de poseer una estructura jerárquica bien definida para cada rol de trabajo.
Para las empresas pequeñas que quieren entregar productos de calidad decente sin acabarse su presupuesto, creemos que definitivamente esta es la mejor estrategia. Sin embargo, para productos de calidad más alta, se recomienda utilizar una estrategia más rigurosa.

4 Bibliografa

Cockburn, A., (2004) *Crystal Clear: A Human-Powered Methodology for Small Teams: A Human-Powered Methodology for Small Teams.* Boston, Addison-Wesley Professional

Agile Alliance, (2015) "12 Principles Behind the Agile Manifesto" en *Agile Alliance.* [En línea.] disponible en:
https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/
[Accesado el día 20 de abril de 2016]