

Métodos Ágiles en el Desarrollo de Software para la Bioinformática

Msc.Maikel Y. Leyva Vázquez¹, Juan Pedro Febles Rodríguez²

¹ Facultad 6, Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana, Cuba.

² Centro Nacional de Bioinformática (BIOINFO), La Habana, Cuba

mleyvaz@uci.cu ; febles@bioinfo.cu

RESUMEN

Los métodos ágiles presentan un enfoque iterativo al desarrollo de software basado en la colaboración y en la automatización de los procesos. Las características de los proyectos de desarrollo de software para la Bioinformática los hace especialmente atractivos para la aplicación de los métodos ágiles. En el presente trabajo se realiza un análisis de estas particularidades y se ofrecen recomendaciones.

Palabras claves: Bioinformática, Métodos Ágiles.

ABSTRACT

Agile is an iterative approach to software development that relies on strong collaboration and automation. Agile methods are especially applicable to Bioinformatics. In this paper recommendations are presented to the application of this methods in an organization.

Key Words: Bioinformatics, Agile Methods.

INTRODUCCIÓN

La Bioinformática (BI) es una disciplina dentro de la Biología, donde las herramientas de la computación tienen una función primordial. Si bien algunos restringen el rango de estudio de la BI al manejo y análisis de bases de datos biológicas —principalmente de secuencias—, podría atribuírsele un sentido más amplio (1), como la fusión de las técnicas computacionales con el entendimiento y apreciación de datos biológicos, el almacenamiento, recuperación, manipulación y correlación de datos procedentes de distintas fuentes.

Esta disciplina se ocupa de almacenar, manipular y analizar grandes masas de información biológica. O sea, trata del uso de las computadoras para el análisis de la información biológica, entendida como la adquisición y consulta de datos, análisis de correlación, extracción y procesamiento de la información. Es un área del espacio que representa la biología computacional, incluye la aplicación de las computadoras y las ciencias de la información en áreas tales como genómica, mapeo, secuenciación y determinación de la secuencia y estructura por métodos clásicos (2).

Una definición más general, la ubicaría como el estudio de la información biológica a partir de la teoría de la información, la computación y las matemáticas. El reto más importante al que se enfrenta esta ciencia es responder a la avalancha de datos que provienen del proyecto "Genoma Humano" y de la genómica (1).

La investigación biológica en general y la BI en particular, precisan que los resultados de las investigaciones en un campo determinado se almacenen en grandes bases de datos regionales o mundiales accesibles para todos los científicos involucrados en la investigación biológica de esa área. La elaboración y validación de herramientas para la BI de manera cooperada constituye otra línea para el trabajo conjunto que debe producir una sinergia importante para la ciencia y la tecnología.

Un área muy importante de la BI se refiere, al desarrollo de instrumentos informáticos para mejorar la investigación biológica. Según Febles y González (2) pueden considerarse como núcleo metodológico de la BI. Algunos de ellos son:

- Instrumentos de software: incluyen amplia variedad de técnicas de software. Algunas de las áreas que cubren dichas técnicas son: Sistemas de Manejo de Información de Laboratorios (LIMS); software para el alineamiento de secuencias y mapas de DNA y proteínas; programas dedicados al análisis de árboles filogenéticos; software creados para el reconocimientos de patrones en proteínas (datamining). Además de estos instrumentos de software se ha realizado un enorme esfuerzo en la construcción, organización, desarrollo y manejo de bases de datos. Otro resultado de la aplicación de software debería ser la construcción de portales de Internet orientados a la presentación y utilización de instrumentos de análisis, listas de discusión, I&D compartido, etc.
- Paralelización computacional: consiste en dividir un gran problema en problemas más pequeños que puedan resolverse en paralelo utilizando diferentes procesadores.

El éxito de la BI depende en gran parte de repositorios donde se almacenan y se hacen disponibles la gran cantidad de datos generados, GeneBank por ejemplo almacena más de 100 gigabases y 61 millones de secuencias biológicas (3). En adición a los repositorios de datos, existe una diversidad de aplicaciones Web (Ej. Blast). También a través de la Web se encuentran disponibles gran cantidad de software bioinformático en sitios tales como Bioinformatics.org o SourceForge, la mayoría de ellos de código abierto.

DESARROLLO

Características de los Proyectos de BI.

En la literatura consultada, los temas relacionados con los métodos de desarrollo del software en la BI son escasos, entre los pocos ejemplos se puede citar (4) (5) (6). Sin embargo, se reconoce la necesidad de mejorar el proceso de desarrollo de software y la calidad del software resultante. A partir de las experiencias personales del autor

relacionadas con proyectos de software bioinformático y de revisiones bibliográficas, identifica las siguientes características de interés en los procesos de desarrollo de software en el dominio en cuestión.

Necesidad de colaboración estrecha. La colaboración reviste especial importancia para la BI (2). En los proyectos se requieren estrechas relaciones de trabajo de al menos dos campos diferentes, la Ingeniería de Software y la Biología. La comunicación entre especialistas de distinta formación representa un reto para el éxito de los proyectos debido a las distintas formaciones, lenguaje de las ciencias etc. Se entorpece notablemente la comunicación, acentuada por la dificultad de transmitir el conocimiento tácito que generalmente poseen los biólogos a los desarrolladores de software.

Equipos pequeños multidisciplinarios. Los equipos de desarrollo de software son generalmente pequeños. Los proyectos de desarrollo de software en BI requieren conocimiento generalmente en Biología, Medicina, Estadística e Ingeniería de Software.

Necesidad de mejorar la calidad del software. Existe necesidad de software de mayor calidad en el campo de la BI, por varios motivos como son el aumento de la confianza en las herramientas que se crean, la necesidad de compartir y expandir el software desarrollado por otros, la necesidad de mejorar la usabilidad de las aplicaciones con el fin de mejorar la productividad de los investigadores, etc.

Uso de herramientas y plataformas libres. Un resultado de la naturaleza colaborativa y la importancia del intercambio en la investigación BI, es el significativo número de programas de código abierto existentes en este dominio (7), y el uso de herramientas y plataformas libres en los procesos de desarrollo.

Cambios frecuentes en los requisitos. Aunque esta característica es común en los procesos de desarrollo de software en varios dominios, en el campo de la BI, donde el software sirve de apoyo a la actividad científica, se puede ver acentuada. La BI en particular y la ciencia en general son actividades exploratorias en las que constantemente se descubren nuevas direcciones en la investigación. Para los desarrolladores que elaboran software para la BI que brinda soporte a estas investigaciones, significa nuevos requisitos. Otro aspecto de interés lo señalan Kane y otros (4) al afirmar que los procesos de desarrollo de software que requieran que los científicos determinen los requisitos detalladamente desde el principio, causan dificultades.

Los Métodos de Desarrollo Ágiles

Los métodos ágiles se desarrollan como un intento por superar las debilidades advertidas y reales en la ingeniería de software convencional (8). Se caracterizan generalmente por su compromiso con los principios fundamentales del “Manifiesto para el Desarrollo Ágil de Software” (9). Aunque el movimiento ágil declaró formalmente su existencia en el año 2000 las prácticas y métodos que se promueven en el manifiesto no son nuevas y tienen una larga historia. Sin embargo la confluencia de nuevas herramientas, libros y cambios en la comunidad de desarrolladores ha creado un gran interés en estos métodos de desarrollo de software en varios dominios entre ellos la BI (10) (11).

Los Métodos Ágiles en los Proyectos de BM

Dadas las características anteriormente citadas de los procesos de desarrollo de software bioinformático, se puede analizar el posible impacto de la aplicación de los métodos de desarrollo ágiles. En la Tabla 2.1 se muestran las

características encontradas y las implicaciones de las prácticas ágiles tradicionales ya sean positivas, negativas o neutras.

Tabla 1 Características de los proyectos de desarrollo de software en la Bioinformática

Características	Implicaciones de la introducción de principios ágiles
1 Equipos multidisciplinarios.	pequeños Positiva. El principio de colaboración presente en el manifiesto ágil resulta relativamente natural. Las características propias de los métodos ágiles los hace especialmente óptimos para equipos pequeños
2 Necesidad de colaboración estrecha	Positiva. El enfoque que utiliza el desarrollo ágil provee un modelo para la colaboración del equipo involucrado en el proceso de desarrollo de software bioinformático.
3 Necesidad de aumentar la calidad del software que se desarrolla y del heredado.	Positiva. Las prácticas ágiles de integración continua, pruebas automáticas, y refabricación hacen factible modificar el software ya existen eliminando los riesgos de afectar características que funcionaban anteriormente aumentando el nivel de calidad que muchas veces falta en proyectos académicos.

Como se puede apreciar las implicaciones son mayoritariamente positivas, sin embargo, es necesario destacar que en algunos contextos se pueden presentar retos para la aplicación de los métodos ágiles. Las organizaciones donde se desarrolla software necesitan, para enfrentarlos, soluciones que no forman parte normalmente de los métodos ágiles.

Los equipos de desarrollo deben responder a esos retos, desarrollando, adaptando o adoptando prácticas adicionales. Cada proyecto tiene sus características particulares que deben ser tenidas en cuenta. Los métodos ágiles no prescriben ningún grupo fijo de prácticas.

El éxito de los métodos ágiles depende de la colaboración de los clientes, en algunos proyectos esto podría representar un reto. Sin embargo, dado el aspecto central que ocupa la colaboración en los métodos ágiles estos pudieran contribuir a que las organizaciones le dieran la atención apropiada al proceso de desarrollo de software. Otro aspecto que pudiera representar un reto son los equipos no co-localizados. Sutherland y otros (12) presentan tres modelos de distribución de equipos y un ejemplo de su aplicación exitosa en un proyecto distribuido geográficamente.

Agilidad vs Formalismo en el Modelo

Boehm y Turner (13) proponen un espacio de 5 dimensiones para determinar que grado de agilidad-formalismo debe tener un proceso de desarrollo de software. También existe una relación inversa entre agilidad y los riesgos de seguridad que enfrente el proyecto. Estos criterios deben ser tenidos en cuenta para instanciar el modelo propuesto.

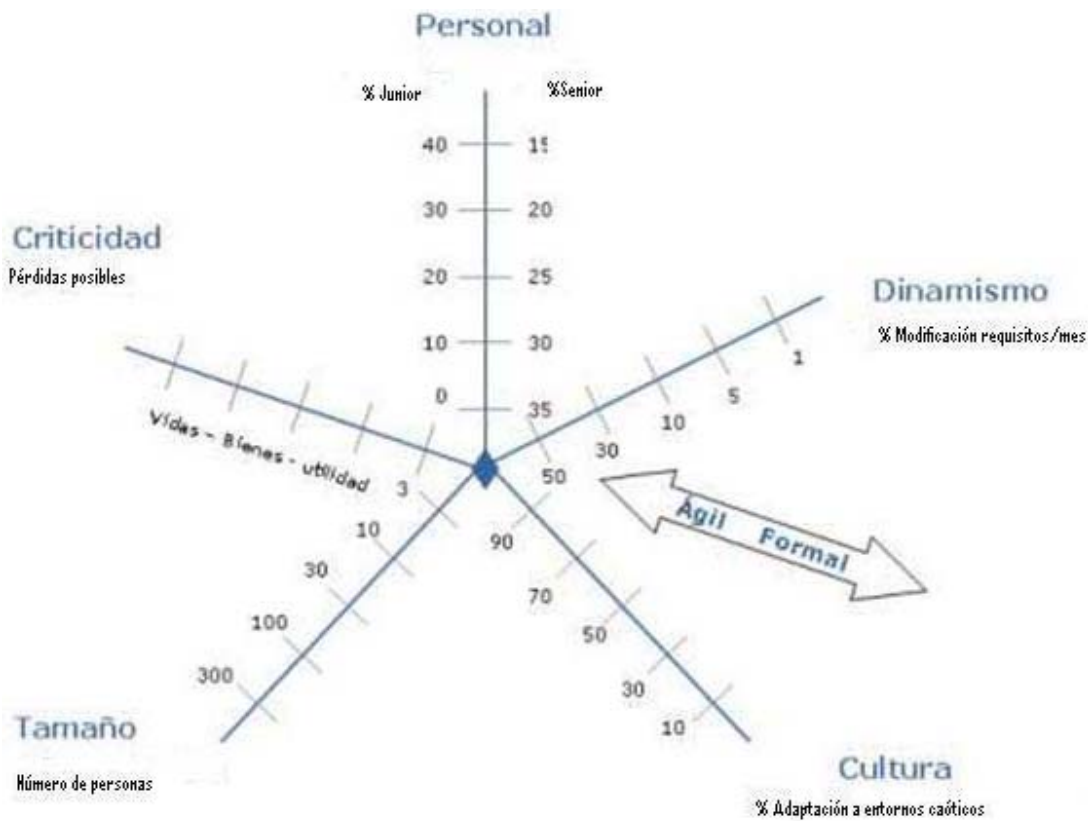


Fig. 1. Estrella Boehm-Turner

Cambio Organizacional

Las organizaciones que deseen mejorar sus capacidades de producir software utilizando métodos ágiles deben, realizar cambios en muchos de los procesos y prácticas actuales. El modelo del proceso de cambio propuesto por Virginia Satir (14) resulta importante para entender y ayudar a que ocurra el cambio en una organización y particularmente las dedicadas al proceso de desarrollo de software.

Para que el modelo propuesto pueda tener éxito es necesario que las organizaciones promuevan la participación activa de todos los interesados, el compromiso con el cambio y motivar el aprendizaje de las prácticas y procesos propuestos. Otro aspecto importante, como señalan Taylor y McGraw (16), es la elaboración de un programa de mejoras y de métricas.

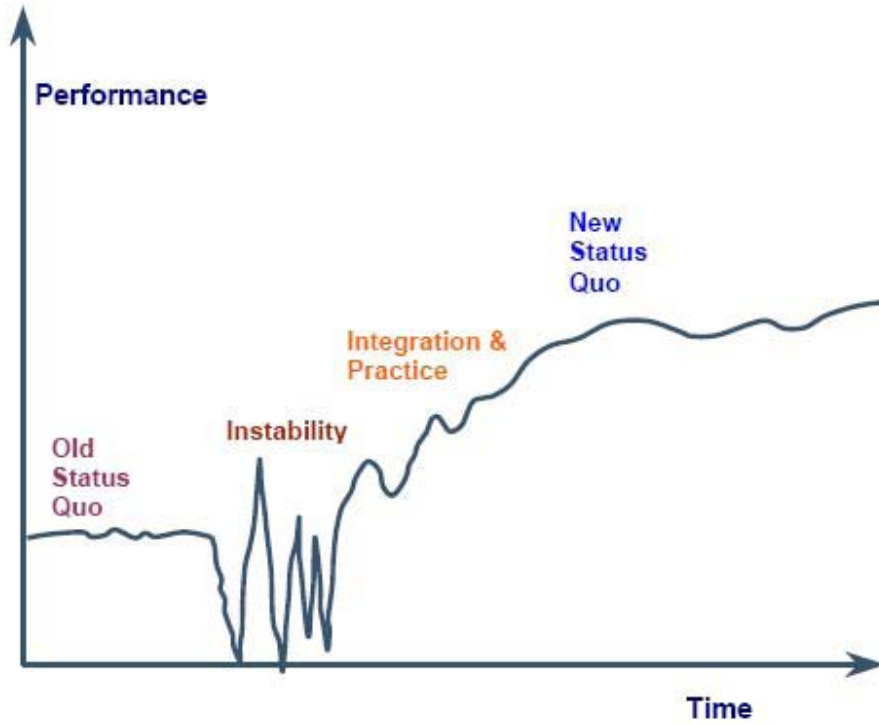


Fig. 2. Ciclo Representativo del Cambio Organizacional (15)

CONCLUSIONES

Los métodos ágiles presentan características que lo hacen especialmente adecuados para la BI. El enfoque que utiliza el desarrollo ágil provee una referencia para la colaboración entre ingenieros de software y otros investigadores en el campo de la Bioinformática.

BIBLIOGRAFÍA

1. **Rivas, Miriam.** Bioinformática. [En línea] [Citado el: 12 de Septiembre de 2006.] <http://www.informaticamedica.org.ar/numero10/monografia2.htm>.
2. *Ciencia de Colaboración.* **Febles Rodríguez, Juan Pedro y González, Abel.** 3, 2002, Vol. 7, págs. 53-57.
3. **NCBI.** NCBI. *GenBank Overview.* [En línea] [Citado el: 20 de Abril de 2007.]
4. *Agile methods in biomedical software development: a multi-site experience report.* **Kane, David W, y otros.** 6 de Enero de 2006, BMC Bioinformatics, Vol. 7. <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1539031>.
5. **Weston, Paul.** *Bioinformatics Software Engineering: Delivering Effective Applications.* s.l. : Wiley, 2004. 0470857722.
6. **Kane, David.** *Introducing Agile Development into Bioinformatics.* [Documento] 2003.
7. **BIOINFOMED.** *Synergy between Medical Informatics and Bioinformatics:Facilitating Genomic Medicine for Future Healthcare.* Madrid : European Commission, 2003.
8. **Pressman, Roger S.** *Ingeniería del Software. Un enfoque práctico.* [ed.] Pablo E. Roig. [trad.] Jesús Elmer Murrieta Murrieta. Sexta Edición. México D.F. : McGraw-Hill, 2006. 970-10-5473-3.
9. **Beck, Kent, y otros.** *Manifesto for Agile Software Development.* 2001. <http://agilemanifesto.org/>.
10. *Agile methods in biomedical software development: a multi-site experience report.* **Kane, David W, y otros.** 5, 6 de Enero de 2006, BMC Bioinformatics, Vol. 7. <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1539031>.
11. **Kane, David.** *Introducing Agile Development into Bioinformatics:.* [Documento] 2003.
12. **Sutherland, Jeff, Viktorov, Anton y Blount, Jack.** *Adaptive Engineering of Large Software Projects with Distributed/Outsourced Teams.* [Documento] 2006.
13. **Boehm, B. y Turner, R.** *Balancing Agility and Discipline: A Guide for the Perplexed.* s.l. : Addison-Wesley, 2003.
14. **Smith, Steven M.** The Satir Change Model. [En línea] 4 de Octubre de 1997. [Citado el: 27 de Abril de 2007.] <http://www.stevenmsmith.com/my-articles/article/thesatir-change-model.html>.
15. **Software Process Subgroup of the Task Force on Security across the Software Development Lifecycle.** *Processes to Produce Secure Software. Towards more Secure Software.* 2004. Vol. I.
16. *Adopting a Software Security Improvement Program.* **Taylor, Dan y McGraw, Gary.** Mayo/Junio, s.l. : IEEE, 2005, IEEE Security & Privacy.