Realizar un nuevo vínculo entre las organizaciones y el software

Gustavo A. Donoso M
gdonoso@inf.udec.cl

La intervención de una organización desde el punto de vista del desarrollo de software a la medida para el apoyo de las actividades que determinan lo que la organización es, no ha resultado ser una actividad trivial en los años en que el software existe como tal. Sucesivas crisis del desarrollo de software han marcado la pauta de una relación conflictiva en ese ámbito, de hecho la tan comentada "crisis" que da origen a la Ingeniería de Software allá por la década de los sesenta aún está presente pese a los esfuerzos que esta última ha desplegado en procura de paliar los efectos de tal crisis.

No se ve la luz al final del túnel para aquellos que intentan, desde distintas perspectivas, superar o por último paliar los efectos negativos que significan estos estados críticos en el desarrollo. Un estudio de U.S. Government Accounting Office que data de 1987, poco más de 20 años luego de las primeras señales de crisis, hace notar que un porcentaje muy pequeño, alrededor del 10%, de los proyectos de software dio resultados satisfactorios que implicaron la utilización del producto tal y como fue entregado. Más del 90% restante de los proyectos fue abandonado, entregados pero no utilizado o pagados pero no entregados [Navon 1992].

Es probable que la situación chilena no varíe mucho al escenario del estudio citado en el párrafo anterior, si bien el estado del arte informático ha cambiado sustancialmente desde aquella época a la actual, tanto que los sistemas operativos, las herramientas de desarrollo y comunicación, las técnicas y metodologías han hecho progresos sustanciales al reducir los esfuerzos destinados al desarrollo, la demanda por software, por otro lado, también a aumentado al hacer más asequibles el hardware a los usuarios finales. Situación que traduce en vanos esfuerzos las mejoras metodológicas y técnicas que se impulsan ya que las propias dinámicas de variación de uno y otro mantienen la diferencia constante.

Por otra parte un aumento de la eficiencia en esas circunstancias no se traduce necesariamente en un aumento de efectividad, realizar más rápido un proceso no significa que aquel se esté haciendo de forma correcta. Esto último es aplicable también al software, en el sentido que automatizar una operación incorrecta lo único que haría es realizar una operación incorrecta diez veces más rápido que antes de su automatización, situación que, al contrario de lo que se esperaba, hace aún más evidente el problema que se buscaba solucionar.

En ese sentido, si bien la eficiencia con que la Ingeniería de Software puede manejar los problemas aporta elementos importantes al momento de enfrentar la crisis, no estaría con ello cumpliendo su objetivo fundacional, superarla. Esta búsqueda de solución orientada sobre la eficiencia de un proceso de producción no es extraña a la condición de los profesionales informáticos ya que en cualquier profesión la búsqueda de soluciones pasa, en una importante medida, por los medios que se dispone para generarlas. Lo que en nuestro caso pasa por la búsqueda de velocidad, nuestra deformación profesional contribuye a que las soluciones que prevemos como adecuadas estén orientadas a la producción de herramientas software que imprimirían la velocidad necesaria a nuestros procesos de producción de manera de que a través de esa aceleración lograr el resultado esperado, es decir, superar de una vez por todas la crisis.

Pero, ello nuevamente nos lleva a formular la pregunta de si estamos, con la Ingeniería de Software, haciendo las cosas correctas. A mi entender no completamente. En la búsqueda de la eficiencia se ha perdido de alguna forma la raíz del problema y, por ello, esta eficiencia que supuestamente se ha logrado no logra su objetivo. En ese sentido nuestra solución es eficiente pero, siento decirlo, no completamente efectiva.

El problema, el cocinero y la amante

A modo de una perfecta novela policial, hemos desarrollado una serie de argumentos y pistas falsas que confunden al lector, paradógicamente nosotros mismos, sobre la verdadera identidad del culpable. Esta argumentación, a mi entender, no se produce realmente por que seamos expertos novelistas que, por razones literarias de una recursividad exquisita, manejemos los hilos del relato de forma que en todo momento tengamos un control consciente de lo que estamos haciendo. En algún punto la conciencia se nos escapa de las manos y el verdadero culpable desaparece de nuestro relato y terminamos acusando, obviamente, al típico mayordomo, infaltable en toda novela policial que se precie. Es nuestro recurso último y desesperado cuando el tiempo apremia y la editorial nos exige la entrega pronta del futuro best seller.

Es obvio en la metáfora anterior que el inocente mayordomo tiene que cargar las culpas de otro y, así, novela a novela se va asentando su sino de chivo expiatorio. Nuestro mayordomo no es otro que el proceso de producción de software, aquel en que la Ingeniería ha gastado los mejores años de su vida.

Pero no todo está perdido, si bien el éxito de nuestras novelas que terminan acusando al inocente -que no lo es tanto al fin- son una buena razón para continuar con esa línea, nuestra pasión literaria debe hacernos reaccionar y, de una vez por todas, aceptar que hay que escribir sin perder el argumento, aceptar la posibilidad de, al menos una vez, desenmascarar al verdadero culpable.

El usuario dirán ustedes. Pobre usuario, otro de nuestros chivos expiatorios. Lamento decirles que pese a que todas las pruebas lo acusan, nuevamente hemos fallado en nuestra apreciación, nuevamente el apremio de no saber como resolver el argumento a jugado en nuestra contra y hemos torcido las líneas de la trama para acusar al inocente -que no lo es tanto.

El culpable se nos escapa, no lo observamos, no tenemos conciencia de su existencia y no hay argumento posible que lo contenga, lo impide el paradigma de nuestra mirada. Somos observadores y como observadores no nos hemos percatado que nuestra observación enreda los argumentos al punto de no darnos cuenta del papel principal que juega nuestro rol en la trama del triller informático. Hemos realizado una puntuación arbitraria de una secuencia ininterrumpida de interacciones [López 1995] y hemos definido con ello nuestro papel fuera del círculo del problema, lo que no es posible.

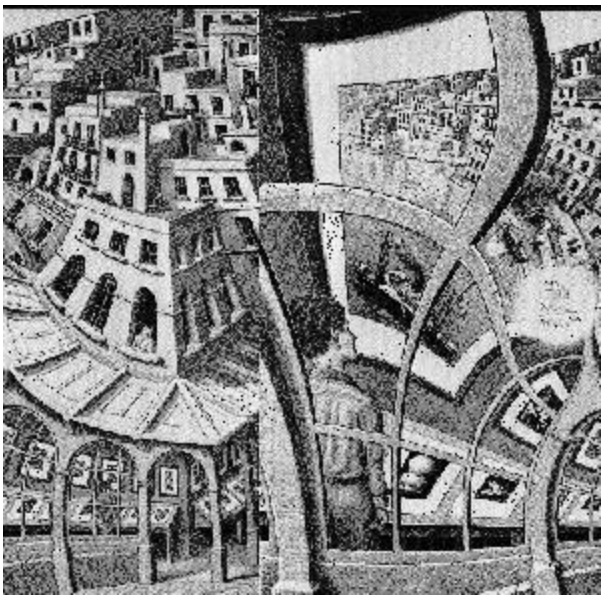
Así el argumento de la novela recae con todo su peso en nuestro rol activo de novelista y culpable. Hemos tratado de desviar la atención en busca de los "verdaderos" culpables que no pueden ser otros que el mayordomo y uno que otro cocinero ocasional para no nombrar a la amante de ambos -noblesse oblige. Pero ha sido en vano, el largo brazo de la ley y los hechos nos llevan nuevamente al principio, al argumento primigenio, al papel nuestro de cada día, al observador que observa sin la conciencia de que su observación cambia lo observado.

Lehmann, Luhmann, y el Ropero

Por allá en el lejano 1980, Meir M. Lehmann escribió un artículo titulado "Programs, Life Cycles, and Laws of Software Evolution" [Lehmann 1980]. En el cual hace una clasificación de los programas (software) según un criterio de complejidad, define tres clases de software de los cuales la última está muy relacionada con la problemática que nos preocupa, dejemos que él mismo nos explique:

> **Programas-E**. La tercera clase, los Programas-E, están inherentemente más inclinados al cambio. Estos son programas que mecanizan una actividad humana o social... La instalación de los programas junto con este sistema asociado -...- cambia la real naturaleza del problema a ser resuelto, el programa puede hasta convertirse en parte del mundo que el mismo modela, está embebido en él...No como otros sistemas artificiales donde,..., el cambio es ocasional, aquí este aparece

> continuamente. La presión del cambio está construida con él.... Meir M. Lehmann, Program, Life Cycles, and..., ACM Proceeding, 1980.



Aquí esta expresada nuestra culpa. Hacemos de nuestro argumento una realidad y a la vez esa realidad se adentra en nuestros argumentos transformado la trama de nuestra novela y nuevamente estamos en un nuevo principio. Somos parte de un cuadro de Escher y nuestra culpa está en que no tenemos conciencia de que estamos en él, formando parte de él como el muchacho de la figura -Galería de Cuadros de M.C. Escher-.Hemos construido una teoría en que la realidad compleja y circular que nos envuelve queda fuera y a modo de la Mecánica Newtoniana nos hemos estrellado con la evidencia empírica de nuestras imposibilidades de ser observadores "objetivos" de aquella realidad.

Nuestras observaciones intervienen el mundo que observamos y con ello le conferimos una dinámica que a su vez interviene en nuestras observaciones y no hay forma de desligarse de aquello. Nuestro rol activo de culpable nos determina y a la ves nos castiga como a Sísifo hijo de Eolo y rey de Corinto condenado a subir una enorme piedra a la cima de una montaña donde volvía a caer sin cesar. Me pregunto si es posible liberar a nuestros modernos Sísifos de su condena eterna.

Hemos construido una teoría en que la realidad compleja y circular que nos envuelve queda fuera y a modo de la Mecánica Newtoniana nos hemos estrellado con la evidencia empírica de nuestras imposibilidades de ser observadores "objetivos" de aquella realidad. Nuestras observaciones intervienen el mundo que observamos y con ello le conferimos una dinámica que a su vez interviene en nuestras observaciones y no hay forma de desligarse de aquello. Nuestro rol activo de culpable nos determina y a la ves nos castiga como a Sísifo hijo de Eolo y rey de Corinto condenado a subir una enorme piedra a la cima de una montaña donde volvía a caer sin cesar. Me pregunto si es posible liberar a nuestros modernos Sísifos de su condena eterna.

Siguiendo con la analogía de Sísifo, la Ingeniería de Software se ha preocupado esencialmente de que Sísifo suba cada vez más rápidamente la piedra a la cima de la montaña, pero no se ha preocupado mayormente que esta no vuelva a caer.

La Ingeniería de Software no está llamada a hacer más duro el castigo, todo lo contrario. Es la esperanza liberadora, el quiebre necesario de la absurda situación. ¿Pero, cómo? se preguntarán ustedes.

La organización no puede ser vista como una máquina. Esta visión simplificante reduciría a las actividades y comunicaciones humanas que sostienen una organización a simples movimientos repetitivos y a incosncientes traspasos de datos entre sus perfectos engranajes. Con ese tipo de modelado organizacional, el problema de Lehmann no existiría y nuestros Sísifos tampoco. No tiene sentido seguirlo y, sin embargo, lo hacemos constantemente, cada vez que pensamos en desarrollar o comprar un software creemos que aquel encajará perfectamente en los engranajes de nuestra organización, como el mejor de los aceites y aquel chirrido desaparecerá para siempre jamás: vanitas vanitatum.

Una organización es un sistema, lo sabemos desde pequeños, ¿pero la vemos como un sistema? o ¿insistimos constantemente en nuestra metáfora de la perfecta máquina? Por entre estas preguntas hay, para mí, una respuesta al castigo, hay una luz de liberación. Pero, vamos por parte: si una organización es un sistema ¿qué tipo de sistema es? ¿qué beneficio puede traer esa buena nueva a nuestros mártires?.

Supongamos, por suponer, que una organización es un sistema social (independiente aún de qué tipo) y veamos que significa aquello. En palabras muy generales un sistema social es un sistema cuyos elementos componentes son personas que realizan el sistema con sus acciones y sus comunicaciones. Una organización productiva, supongamos una empresa, es un sistema social con un propósito definido, propósito que condiciona qué acciones y qué comunicaciones son factibles de realizar por las personas que componen ese sistema social y, si somos sediciosos, podemos pensar que las acciones y comunicaciones que las personas realizan en el seno de una organización productiva determinan lo que ésta es. Muchas empresas que tienen similares objetivos -por no decir iguales- son muy distintas en su realización.

Pero si queremos tener una visión más profunda de lo que es un Sistema Social, es posible recurrir a las últimas teorías en sociología. Según Niklas Luhmann [Rodríguez 1991] un sistema social se encuentra compuesto de comunicaciones que generan comunicaciones, donde las comunicaciones componen una red de producción de comunicaciones que pasan a formar parte de aquella red, conformando una unidad cerrada y autorreferente, esta referencia a sí mismos se entiende en que no reciben comunicaciones desde el exterior ni entregan comunicaciones a su entorno. Por lo cual, para que algo que pertenece al entorno del sistema, pase a formar parte de él, debe ser tematizado por éste, transformándose en una comunicación del sistema, esta tematización es posible gracias al "sentido" que sería una estrategia selectiva que mantiene en suspenso las posibilidades comunicacionales no actualizadas, lo que permite tender un puente entre las comunicaciones que componen temporalmente al sistema [Rodríguez 1991, p.114]. El sentido constituye al sistema social, pero también es constituido por éste. Así, para Luhmann un sistema social es un sistema autopoiético de comunicaciones que surge y se mantiene gracias a su propia dinámica.

De forma similar, una organización, para Luhmann, es un sistema que se compone de decisiones, y que elaboran las decisiones de las cuales se componen a través de sus decisiones componentes. Es decir, simplificadamente es un sistema autopoiético de decisiones.

El concepto de autopoiesis fue elaborado por Humberto Maturana y Francisco Varela para explicar la organización de los seres vivos. Luhmann encuentra en él el punto de apoyo a su teoría sobre los sistemas sociales, generalizándolo; lo que determina diferencias que no resultan del todo felices desde la perspectiva que originalmente movía a sus creadores. No es posible en este artículo profundizar sobre los conceptos y diferencias de ambas teorías y se recomienda al lector interesado hacerlo, ya que la riqueza de ambos enfoques permite ampliar sustancialmente los dominios de explicación sobre los seres vivos y, especialmente, sobre las organizaciones sociales.

Volviendo a nuestra problemática original, esta perspectiva de las organizaciones provee elementos sorprendentemente claros para la validación de la tesis de Lehmann. Al ser el sistema organizacional un sistema autopoiético de decisiones tenemos que sus invariancias son producto de la actualización constante de las decisiones que lo componen y en ese sentido la dinámica autopoiética es una constante que permite la estabilidad del sistema como organización, es decir, la identidad de la organización existe producto ya no de invariancias estructurales sino como resultado de una dinámica que reconstruye su estructura a cada instante, desde esta perspectiva el cambio es algo connatural a los sistemas organizacionales y es el cambio lo único que permanecería como constante, una organización así es una que se reconstruye contantemente a sí misma y cuyo único fin es esta reconstrucción.

Así, lo que ocurriría al desarrollar -o comprar- software es que para que éste sea internalizado como

una comunicación válida, que corresponde con el sentido de la organización, éste debe ser tematizado como una comunicación que realiza la autopoiésis de la organización, pero esta tematización trae como consecuencia una variación del sentido de la organización, lo que invalidaría, probablemente, algunas de las comunicaciones que componen actualmente la organización, lo que a su vez, probablemente, varíe nuevamente el sentido de la organización y así hasta llegar a un nuevo estado homeostático que realice la nueva identidad organizacional. La estabilidad que lograría la organización con este nuevo estado debe incorporar al software como una comunicación tematizada por el sentido organizacional, el software al ser desarrollado -o comprado- en base a un modelo esperado de organización debería llevar a ésta hacia aquel modelo.

Lo que ocurre es que esta deriva esperada generalmente presupone condiciones que no son tan reales, de hecho los modelos de organización que se utilizan son más bien mecanicistas y no sistémicos, lo que implicaría que los resultados esperados -desde la perspectiva mecanicista- sean distintos de los realmente obtenidos -explicables desde una óptica sistémica. Y es esto lo que produce la caída de la piedra desde la cima de la montaña.

Así, a modo de conclusión, es nuestra visión de la organización la que debemos mejorar a través y para la Ingeniería de Software, debemos considerar nuestra posición de observadores y nuevos modelos de organización que eviten las visiones reduccionistas a las cuales los antiguos paradigmas nos tienen habituados.

Hagamos como Clive S. Lewis que lleva a los jóvenes protagonistas a través de un ropero a la tierra de Narnia, donde el dios-león, Ashlan, la reconstruye constantemente con su canto.

Bibliografía

- [López 1995] López, A.; Parada, A.; Simonetti, F. "Introducción a la psicología de la comunicación" Ediciones Universidad Católica de Chile, 1995
- [Lehmann 1980] Meir M. Lehmann, Program, Life Cycles, and..., ACM Proceeding, 1980
- [Navon 1992] Navon, J.; Fuller, D.; Casas, I. "Fundamentos y técnicas modernas de programación", Ediciones Universidad Católica de Chile, Santiago, 1992
- [Rodríguez 1991] Rodríguez, D.; Arnold M.; "Sociedad y Teoría de Sistemas", Editorial Universitaria, Santiago, 1991.