

José Miguel Santibáñez Allendes*

RESUMEN

Considerando las diversas explicaciones que se tienen de las metodologías de desarrollo de Sistemas de Información (cada autor, propone sus propias variantes), el objetivo de este trabajo es ayudar a ordenar la discusión, entregando definiciones que guíen la conversación y faciliten la crítica objetiva y la proposición de alternativas a metodologías, métodos, herramientas y procedimientos existentes hoy en día o en el futuro.

1.- INTRODUCCIÓN

Al analizar las diferentes fuentes bibliográficas disponibles en las áreas de Sistemas de Información e Ingeniería de Software, una de las primeras conclusiones importantes, es que no existe una propuesta que sea común a los distintos autores. No sólo por la existencia de metodologías distintas (por ejemplo la “estructurada” o la “orientada a objetos”) dos autores distintos (o incluso el mismo autor en distintas ediciones de su libro) tratan una misma metodología de maneras diferentes.

Algunos autores, han optado por hacer una descripción detallada de distintos modelos que se pueden aplicar en un desarrollo de software, otros plantean su propia visión de cómo se debe aplicar determinada metodología, llegando al punto, de ignorar (o rechazar) la existencia de alternativas a lo que proponen. Por si ello no bastara, cada profesional del área, tiene su propia opinión de qué herramientas o modelos tienen resultados y son útiles de aplicar y cuáles no (y dentro de ellos, con múltiples variaciones de cómo, cuándo y dónde aplicarlos); y eso, descontando a esa gran masa que considera que el desarrollo de software es un arte, que ellos mismos son artistas y que cualquier documentación o metodología en el desarrollo del software, es una tranca a la creatividad y, en buen chileno, un “cacho” que, de ser necesario, se le asigna al más nuevo de los contratados, como “prueba de fuego” o “bautismo” que debe superar.

* Ingeniero Civil en Informática, USACH. Académico UCINF.

Intentar compilar toda la información relativa a la Ingeniería de Software, es una tarea titánica que, probablemente, no daría frutos. Más aún, cuando quienes vengan en el futuro, tendrán tanto derecho como quienes hoy están desarrollando software, para criticar las metodologías existentes y proponer sus propias modificaciones o mejoras. Sin embargo, el sentar algunas bases sólidas que ayuden a encauzar la discusión de Metodologías, es una tarea prioritaria hoy en día.

El punto de partida para ello, es rescatar las definiciones adecuadas, aquellas que pese a existir, son habitualmente ignoradas por los ingenieros. Luego, en función de esas definiciones, es posible construir los términos que deben ser utilizados en la discusión, crítica y nuevas proposiciones sobre el tema.

El presente documento, pretende convertirse en un canal de discusión que permita alcanzar el consenso respecto de los conceptos asociados a la palabra “metodología” y su aplicación en el ámbito de la Ingeniería Informática.

Una reflexión inicial y necesaria, surge de constatar que, mientras la mayoría de las ramas de la ingeniería han dispuesto de muchos años para establecer y comprobar sus teorías y prácticas, la Ingeniería de Software es un verdadero recién nacido y que a la fecha, como “ingeniería” no tiene más de 20 años.

Al mirar la construcción de un edificio, sorprende la sincronización con la que actúan los distintos operarios. Los camiones llegan a descargar material poco antes de que sean necesarios y retiran las sobras en el momento indicado. Las grúas, desde “plumas” a “bobcats” llegan y permanecen en escena el tiempo necesario y son instalados y retirados con tal celeridad, que parece producto de la magia.

Más de un cliente ha esperado eso del desarrollo de sus sistemas. Que analistas y programadores, cual ballet con años de práctica, se sincronicen y funcionen como reloj, logrando sistemas en tiempos mínimos, casi “just in time”. Por supuesto, esa esperanza ignora que la Ingeniería Civil, encargada de esas construcciones, lleva miles de años de práctica, muchos más que los de caminos y puentes del imperio romano, e incluso más que los de las pirámides egipcias.

2. DEFINICIONES

El punto de partida de cualquier conversación, debe ser el establecer el conjunto de términos comunes que serán utilizados en la discusión. Quizá la mayor fuente de discrepancias, se debe a que los términos usados son entendidos de distintas maneras por los participantes.

Este proceso, se inicia con la palabra que sirve de título a este artículo y sigue con todos aquellos términos que aparecen en las mismas definiciones. Si bien inicialmente se propone una definición en función de la experiencia del autor, se han buscado otras fuentes que den soporte a la acepción dada.

Metodología: Colección de métodos de solución de problemas organizados bajo una filosofía común y gobernados por un conjunto de principios.

Según la RAE, se define la palabra metodología como:

1. Ciencia del método.
2. Conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal.

Habitualmente, la palabra metodología acostumbra a ser utilizada según la segunda acepción y, a falta de otra palabra, se propone mantener esa definición.

Método: Forma de hacer las cosas.

Según la RAE, se define la palabra método como:

1. Modo de decir o hacer con orden una cosa.
2. Modo de obrar o proceder; hábito o costumbre que cada uno tiene y observa.
3. Procedimiento que se sigue en las ciencias para hallar la verdad y enseñarla. Puede ser analítico o sintético.
4. Obra que enseña los elementos de una ciencia o arte.

Pressman, por su parte, indica que “Los *métodos* de la ingeniería del software indican “cómo” construir técnicamente el software”. (Pressman, 1993).

Se entiende por Método, al modo de hacer las cosas, que le indica a una persona qué debe hacer a continuación y que permite que, de ser necesario, se pueda remplazar a esa persona en medio de un trabajo, sin que ello implique un atraso en el proyecto.

Técnica: Un conjunto de procedimientos precisamente descritos para lograr una tarea estándar.

Nuevamente según la RAE:

1. Conjunto de procedimientos y recursos de que se sirve una ciencia o un arte.
2. Pericia o habilidad para usar de esos procedimientos y recursos.
3. Habilidad para ejecutar cualquier cosa, o para conseguir algo.

Es importante notar la diferencia entre “hacer algo” de la definición de la RAE, respecto de “lograr una tarea estándar” de la definición inicial. El problema con la definición de la RAE, es que “algo” es absolutamente ambiguo y puede significar desde la aplicación de una herramienta, hasta el desarrollo de un proyecto complejo. Considerando el objetivo inicial de establecer un consenso respecto del conjunto de términos aplicados, es necesario evitar cualquier ambigüedad, aún cuando eso signifique reducir el alcance del término original.

De esta manera, la definición se cenura en una “tarea estándar”, una acción específica, claramente definida y acotada, que permite alcanzar un objetivo muy específico, y que se puede lograr utilizando un conjunto acotado de herramientas.

Herramienta: Instrumentos o ayudas tangibles en la realización de una tarea.

Según la RAE:

1. Instrumento, por lo común de hierro o acero, con el que trabajan los artesanos.
 2. Conjunto de estos instrumentos.
 3. Máquina herramienta.
 4. Herraje l.
 5. Fam. Arma blanca, puñal, navaja, faca.
 6. Fig. y fam. Cuernos de algunos animales, como el toro y el ciervo.
 7. Fig. y fam. Los dientes de la boca de una persona o un animal.
-

Aún cuando en el castellano, las palabras “Herramienta” e “Instrumento”, no son exactamente intercambiables¹, debido a un comprensible error de interpretación en la traducción, la palabra inglesa “Tools”² hace pensar más en “Herramientas” que en “Instrumentos”.

Orientado hacia el objetivo principal de este documento, se entrega la definición de la RAE del término “instrumento”:

1. Conjunto de diversas piezas combinadas adecuadamente para que sirva con determinado objeto en el ejercicio de las artes y oficios.
2. Ingenio o máquina.
3. Aquello de que nos servimos para hacer una cosa.
4. Instrumento músico.
5. Lo que sirve de medio para hacer una cosa o conseguir un fin.
6. Escritura, papel o documento con que se justifica o prueba alguna cosa.

Al observar las acepciones 1, 2, 3 y 5 es claro que cuando en ingeniería se usa el término “Herramienta”, se hace referencia a “Instrumento”.

Según Pressman, “Las *herramientas* de la ingeniería del software suministran un soporte automático o semiautomático para los métodos” En esta categoría, Pressman se refiere principalmente, al uso de herramientas de software, como por ejemplo, productos “CASE” (ingeniería de software asistida por computador, sigla en inglés). (Pressman, 1993).

Herramienta de Software: Un paquete de programas para computadores para asistir en una o más técnicas de una metodología. La función de las Herramientas de Software es facilitar el trabajo de las personas involucradas en el proyecto.

 1 La palabra «Herramienta» tiene una clara connotación artesanal, mientras que la palabra «Instrumento» tiene mayor connotación científica, aunque también musical.

2 Con mayor rigurosidad, es conveniente decir que la palabra «tools» tiene como mejor traducción a la palabra «utensilio», la cual, según la RAE se define como: «Lo que sirve para el uso manual y frecuente». Y también: «Herramienta o instrumento de un oficio o arte».

Procedimiento: Cómo poner en práctica las herramientas.

Según la RAE:

- 1.m. Acción de proceder.
- 2.Método de ejecutar algunas cosas.

Siendo Proceder:

- 1.m. Modo, forma y orden de portarse y gobernar uno sus acciones bien o mal.

Y según Pressman: Los *procedimientos* de la ingeniería del software son el pegamento que junta los métodos y las herramientas y facilita un desarrollo racional y oportuno del software de computadora". (Pressman, 1993).

3. FUNDAMENTOS DE LA INGENIERÍA

Siguiendo la línea de las definiciones, es útil señalar que según la RAE **Ingeniería** es el "Conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización de la materia y de las fuentes de energía".

En lo principal, la Ingeniería se basa en la utilización de métodos, con pasos organizados y repetibles. La tradición de la ingeniería es la de seleccionar un conjunto de pasos o etapas que de una u otra manera pueden ser agrupados en cuatro secciones fundamentales y claramente definidas:

- **Análisis:** es siempre el inicio del trabajo, no se empieza nada sin un estudio previo de la situación. El análisis apropiado en la tradición de la ingeniería debe dejar documentación apropiada para que cualquier otro ingeniero conocedor de la metodología sea capaz de tomar el testimonio (cual carrera de postas) y continuar SIN pérdida de trabajo.
- **Diseño:** Concluido el análisis, es necesario decidir qué y cómo se va a dar solución al problema planteado. Al igual que el análisis, el diseño debe producir resultados tales que permitan el reemplazo del ingeniero que está trabajando en cualquier momento. Sin diseño NO hay Ingeniería.

- **Construcción:** Una vez completados (al menos parcialmente) los pasos anteriores, se puede empezar a traducir los aspectos diseñados en una solución real (software + archivos/bases de datos)

- **Pruebas:** La última parte antes de poder entregar la solución al usuario, para su uso en el medio. Y aunque no hay disponible una definición precisa de "Ingeniería del Software", Pressman rescata: "El establecimiento y uso de principios de ingeniería robustos, orientados a obtener software económico que sea fiable y funcione de manera eficiente sobre máquinas reales". (Pressman, 1993).

Queda también la noción que se puede obtener al extender la definición de la RAE para el término Ingeniería, entendiendo que la Ingeniería de Software es el que el "Conjunto de Conocimientos y Técnicas que permiten el desarrollo de software", en otras palabras, las Metodologías de desarrollo de Software.

4. ELEMENTOS IMPORTANTES

Toda Metodología, considera a lo menos cuatro elementos importantes:

Principio Rector: También denominado "filosofía de la metodología", es la norma o idea fundamental que rige el pensamiento o la conducta, y orienta el análisis, diseño y desarrollo del software. Es el Principio, el que ordena y estructura las herramientas que son aplicables en la metodología, así como los Procedimientos con los que se aplica. Tradicionalmente, se apellida a cada metodología en función del principio que la rige:

- "Metodología estructurada"¹ se fundamenta en que lo más importante de un sistema de información, son las estructuras que lo componen y que, por lo tanto, el análisis se debe centrar en ellas, descomponiéndolas en nuevas subestructuras hasta tener elementos tan simples, que puedan ser resueltos en forma sencilla.

- "Metodología orientada a objetos" indica que el principio rector es la orientación a objetos, es decir el análisis de todos los componentes del sistema como un conjunto de objetos que poseen propiedades y que, a través de mensajes, se interrelacionan entre sí.

¹ En más de una oportunidad, diversos profesores del área, se han referido a ella como "Metodología Clásica, pues es producto de la evolución histórica del desarrollo de software.

Herramientas: son definiciones de mecanismos manuales, semiautomáticos o automáticos que permiten analizar, diseñar o construir el software. Las herramientas quedan estrechamente ligadas al principio rector de la metodología y es muy poco probable que una misma herramienta sea utilizable en más de una metodología¹. Una herramienta debe tener un objetivo específico y un método de aplicación. Por lo general, se ha demostrado que las herramientas gráficas (que usan imágenes) son más fáciles de usar y entender que las herramientas que sólo se sustentan en textos escritos. Son ejemplos de herramientas: los DFD, MER, Lenguaje Estructurado, Diagramas de Componentes, Diagramas de Herencia, etc.

Procedimientos²: Se refiere al modo de hacer, con orden, las cosas; es decir, cómo poner en práctica las herramientas. Los procedimientos corresponden a la definición que permite unir y ordenar los resultados de cada herramienta y facilitan el desarrollo racional y oportuno de software. Definen la secuencia en la que se aplican las herramientas, la entrega de los resultados de ellas, los controles que ayudan a asegurar la calidad. También coordinan y controlan los cambios y entregan las directrices que ayudan a los administradores a evaluar el progreso del proyecto.

Modelos: El modelo define las etapas a realizar para alcanzar la solución al problema planteado. Los Modelos, se refieren a la forma de organizar los Procedimientos, de manera de obtener resultados de calidad en el menor tiempo posible. A diferencia de las Herramientas y los Procedimientos, los modelos son relativamente independientes del principio, pudiendo aplicarse sin grandes dificultades, cualquier modelo a cualquier metodología. Pese a lo anterior, el modelo debe quedar definido claramente antes de iniciar el desarrollo del software. Ejemplos de modelos son: Cascada, Prototipos, Espiral, T4G, RAD:

- **Cascada:** También denominado “clásico” (Pressman, 1993; Barros, 1990). Bajo este modelo, los procedimientos de la metodología se ordenan en pasos o etapas, las cuales deberán ser seguidas bajo un enfoque secuencial de análisis, diseño y desarrollo. Creado a partir del modelo convencional de “línea de producción” de la ingeniería clásica, este modelo es el más aplicado en el desarrollo de Software.

¹ A menos que existan elementos comunes en la definición del principio rector de cada una de las metodologías.

² Se ha optado por la palabra “Procedimiento” para prevenir el conflicto entre las palabras “Método” y “Metodología”. sin embargo, tanto en la definición como en el uso, se observa claramente que son términos intercambiables.

- **Prototipos¹**: Los prototipos son modelos (no necesariamente productos de software) que permiten estudiar y probar aspectos específicos del producto final (en este caso el producto de software). Bajo este modelo, se planifica la aplicación de las diferentes herramientas, para producir elementos de pruebas específicas (interfaz de usuario, mantenedores, procesos) que deberán ser presentados al usuario y confirmados por éste.

Alternativamente, se ha denominado de esta forma, al resultado del diseño rápido de productos de software que permitan comprender de mejor manera los requerimientos del usuario. Sin embargo, para prevenir confusiones, se sugiere que para esos casos, se usen las denominaciones siguientes, según corresponda.

- **Espiral**: El modelo espiral, pretende optimizar los tiempos y reducir la incertidumbre del proyecto, así, la idea es partir produciendo una pequeña parte del sistema (pero completamente funcional) y una vez completada, se procede a crear una segunda parte, acoplada a la primera, de manera que en cada iteración, se obtiene una versión aumentada del sistema. El proceso concluye cuando se considera que el sistema ha alcanzado un nivel de maduración tal, que permite que el trabajo para el que fue creado, sea realizado sin mayores inconvenientes.
- **T4G o RAD(D)**: T4G es la sigla de “Técnicas de 4^ª Generación” y RAD(D) es la sigla de “Rapid Application Development (and Deploy)” o “Desarrollo (y Distribución) rápido de aplicaciones”. Como modelo, se basa en la existencia de herramientas de software que se caracterizan como “T4G” y “RAD(D)”, las cuales permiten que el analista diseñador de un sistema, realice un mínimo análisis y diseño, lo traduzca rápidamente en aplicación y se lo presente al usuario para su estudio y posterior aprobación o indicaciones para modificación.

Actualmente, éste es, con una alta probabilidad, el modelo más utilizado por los desarrolladores de software; sin embargo, y probablemente en la misma tasa de ocurrencia, es llamado “modelo prototipo”.

¹ No está de más la definición RAE: “Ejemplar original o primer molde en que se fabrica una figura u otra cosa”.

5. A MODO DE REFLEXIÓN

Quizá llame la atención del lector, que en este documento no se han definido claramente las etapas que ordena cada modelo. Ni siquiera se ha hecho un listado de herramientas o procedimientos según alguna metodología. La razón es simple, el autor de este texto está convencido de que NO existe una definición estricta ni que sea posible siquiera intentar un consenso al respecto. Al revisar literatura especializada, queda claro que cada autor (incluyendo al suscrito) tiene sus propias versiones al respecto de cómo y cuándo utilizar cada herramienta, o la división del proyecto en etapas para dar solución a un problema específico.

Esto es positivo. La historia enseña que en la Ingeniería Informática se han cometido numerosos errores, y se debe aprender de ellos. La lección principal, es que aún hoy, existen jefes de proyectos, analistas, diseñadores y/o programadores (mención a cargos, no títulos) que al enfrentar un proyecto, actúan más como “artistas” inspirados por la divina providencia que como personas metódicas. Y eso ocurre porque muchas veces llega un jefe de proyecto “inspirado” y decide poner en práctica toda una serie de elementos metodológicos aprendidos en el último seminario al que asistió, sin tener claro si ellos son aplicables en la realidad de su empresa o no.

La metodología, para que sirva, debe cumplir con dos condiciones fundamentales:

- 1. Debe tener Hitos bien definidos:** El analista/diseñador debe saber claramente cuáles son los objetivos de la etapa en la que se encuentra, reconociendo claramente las tareas que debe realizar para alcanzar dichos objetivos.
- 2. Debe ser Incremental:** El resultado de una etapa, debe ser de utilidad para la persona que va a realizar la etapa siguiente.

Frecuentemente, se escucha a diversas personas del área informática, reclamando amargamente por el hecho de tener que crear un documento que refleje, por escrito el diseño del trabajo realizado. En esos casos, se tiene un proyecto cuya metodología de desarrollo no cumplió con las condiciones arriba señaladas. Si la metodología define pasos que no son de utilidad al desarrollador, entonces éste lo dejará para el último momento, rompiendo así con la idea fundamental de tener una metodología.

No está de más, terminar esta reflexión, recordando que una metodología, sea ésta la que sea, debe constituir un estándar de desarrollo. Y los estándares no son tales por ser buenos, bonitos o baratos, sólo son estándares, cuando son utilizados como tales.

BIBLIOGRAFÍA

Barros V., Oscar (1990). **Manual de diseño lógico de sistemas de información administrativos**, Santiago, Editorial Universitaria.

Kendall, Kenneth E.;Kendall, Julie E. (1991). **Análisis y Diseño de Sistemas**. Madrid, Prentice-hall Hispanoamericana.

Pressman, Roger S. (1993) Ingeniería del Software, un enfoque práctico. 3ª. ed. Madrid, McGraw-Hill/Interamericana de España S.A.

Real Academia Española (1995). Diccionario de la Lengua Española, Edición Electrónica, versión 21.1.0, Espasa Calpe S.A.

Sommerville, Ian (1988) Ingeniería de software, **Sistemas Técnicos de Edición**.

Yourdon, Edward. (1993). **"Análisis Estructurado Moderno"**, Prentice-Hall Hispanoamericana.