José Teodosio Lorente Vallecillos 77390343N

Relación 1.1

Ejercicios:

(2)

Existen tres tipos de software, el software de ~~sistemas~~ código abierto, el software de ~~aplicaciones~~ código cerrado y el software de ~~programación~~ dominio publico.

El software de dominio publico puede no ser clasificado como licencia, ya que basicamente, se establece "este adjetivo se da a todo aquel software que es distribuido sin ningun tipo de restricción" hacia el usuario final. Por lo tanto su codigo puede ser modificado, usado y distribuido por cualquiera; y ademas, los mismos usuarios tienen la capacidad de licenciar y cambian el tipo de software de sus redistribuciones libremente a como mejor les convenga. Y un ejemplo de uso es UNIX.

El software de código abierto es el termio con el cual se identifica al software distribuido y desarrollado bajo una licencia que permite a los usuarios el acceso al código fuente del software, y que de igual manera puede ser estudiado y modificado sin ninguna restricción en el uso del mismo, con la posibilidad de poder redistribuirlo, siempre y cuando, sea bajo los términos y condiciones de la licencia con la cual fue adquirida el software original. Y un ejemplo de uso es LINUX.

El software de codigo cerrado se trata de una clasificación donde el autor del proyecto limita los derechos de copia, modificación y redistribución de su proyecto. Y en caso de que el usuario final desee llevar a cabo cualquiera de estas acciones, requiere de pagar una cierta cantidad al autor para tener derecho a hacerlo. Y un ejemplo es el ~~los~~ SO de Windows.

(4)

1er ejemplo de mito = si un proyecto se retrasa conforme los tiempos marcados, se puede arreglar la diferencia de progreso añadiendo más programadores, es decir, si contratas a más trabajadores, el proceso se acelera.

1º realidad = El desarrollo del software no es un proceso mecánico similar a la manufactura o a la industria automovilística, es mas a medida que se agregan personas, las que ya (~~ya~~) se encontraban trabajando deben dedicar tiempo para enseñar a los recién llegados, lo que disminuye el tiempo dedicado a continuar el (~~progreso~~) desarrollo. Es decir, más programadores, conlleva a una menor productividad, pero en funcion de la cantidad y la coordinación planeada.

José Teodosio Lorente Vallecillos 77390343N

(4)

2º ejemplo de mito = Lo unico que se entrega del trabajo producido en un proyecto exitoso es solo el programa que funciona.

2ª realidad = Un programa funciona porque tiene varias partes necesarias para su configuración, modelos, documentos, planes; son la base de una buena ingeniería, y quien el apoyo para el software.

(5)

Este principio nos dice que pensemos bien en cada situación profesional, ya sea rechazando sobornos, mantener objetividad profesional, mantener valores humanos, etc; y un ejemplo es que te pidan hacer un software ~~para que con~~ basado en inteligencia artificial, que se dedique a reconocer, fijar y destruir objetivos, da igual cuales sean, eso es inmoral y algo que se debe rechazar.

(6) El software evoluciona generando versiones cada vez más complejas, eventualmente al dejar de evolucionar, cumplirá su ~~ciclo~~ ciclo de vida, entrará en obsolescencia y será reemplazado por un producto nuevo.

(7) Desarrollo de Aplicaciones de Inteligencia Artificial

El proceso de desarrollar y mantener un producto software es llamado ingeniería del software. Este proceso atraviesa varias fases, que en su conjunto se llaman ciclo de vida. Todas las actividades de estas fases son necesarias para definir, diseñar, desarrollar, probar, operar y mantener una aplicación. ~~Ha~~ Hay muchos modelos diferentes y hay variantes de estos modelos que enfatizan en diferentes aspectos del ciclo de vida del software y que son apropiados para un amplio rango de situaciones. Tres ~~modelos~~ de estos modelos son el modelo de cascada, el modelo de prototipos y el modelo en espiral de Boehm.

- Modelo de cascada. Es un paradigma de ingeniería de software convencial, en el cual el ciclo de vida está muy estructurado. En cada fase, el resultado tiene que ser construido y verificado correctamente. Por correcto se entiende que la aplicación cumple las especificaciones y los requisitos. La cascada es un modelo secuencial dirigido hacia los últimos niveles del ciclo de vida. Cuando se desarrollan sistemas basados en conocimientos el experto está involucrado en todas las etapas de desarrollo. Esta implicación constante resulta en un cambio dinámico de las especificaciones y los requisitos, en contraste con el desarrollo de software convencional, donde los requisitos son esencialmente estáticos. Por ello el modelo de cascada no es lo más apropiado para el desarrollo de una aplicación AI. El cambio dinámico de las especificaciones conlleva que los ingenieros tengan que revisar la fase anterior constantemente.

José Teodosio Lorente Vallecillos 77390343N

(7)

- Modelo de prototipos. Es un paradigma de ingeniería de software con múltiples iteraciones, a través del cual se llevan a cabo las fases de adquisición de conocimientos, de codificación y prueba hasta que se obtiene un prototipo final de una aplicación AI. Este modelo tiene un grave problema, y es que la complejidad de la construcción, el tiempo y los costes de los prototipos no se escalan de forma lineal con la dificultad de codificar prototipos parciales. Consecuentemente este modelo de desarrollo se limita a problemas pequeños, o a partes de modelos más complejos; por ejemplo, con este modelo no se podría realizar el ~~desarrollo~~ problema de la conducción autónoma en el sector automovilístico.
- Modelo en espiral de Boehn. Consiste en un refinamiento del modelo de cascada integrando el modelo de prototipos. En este modelo las fases análisis de requisitos y diseño global son iguales que en el modelo de cascada. Luego se dividen las fases que implican el diseño detallado y la implementación. Una característica importante de este modelo es que permite añadir nuevos requisitos en el proceso de desarrollo. Así se facilita el desarrollo dinámico de las especificaciones del sistema.

Teniendo en cuenta lo mencionado anteriormente, se llega a la conclusión de que el modelo de proceso que mejor se adapta al proyecto es el modelo en espiral de Boehn.

Una vez seleccionado el modelo de ingeniería de software, espiral de Boehn, el desarrollo de una aplicación de Inteligencia Artificial debe ser dividido en las siguientes fases:

1. Valoración. En esta primera fase, las acciones a seguir son seleccionar los objetivos, analizar los requisitos, estimar el potencial y seleccionar el equipo de desarrollo, el tiempo empleado en esta fase es: 2 días, (48 horas).

2. Especificación del sistema. En esta fase, hay que crear un modelo conceptual de la aplicación, diseñar un modelo de la aplicación, seleccionar las técnicas de Inteligencia Artificial apropiadas y formalizar un prototipo inicial. El primer prototipo, como resultado de esta fase, debe centrarse en una descripción general de las futuras funcionalidades del sistema, en su comportamiento y en su estructura, el tiempo empleado en esta fase es: 400 horas.

3. Refinamiento e implementación. Esta fase implica el desarrollo de una aplicación detallada. ~~(por ejemplo, en un modelo de desarrollo incremental)~~ Esta fase implica seleccionar los sucesivos incrementos que forman una extensión funcional del modelo de diseño. Cada futuro incremento tiene que ser codificado y probado. Hay que tener en cuenta que todos los incrementos deben ser totalmente compatibles con los desarrollos previos y futuros. Cuando el último incremento se ha implementado, el conjunto de la especificación debe sentar completo como lo sería en un modelo en cascada. Todos los subsistemas pueden integrarse y ser probados como un conjunto. Tiempo empleado = 6 meses.

Jose Teodosio Lorente Vallecillos 77390343 N

(7)

4. Sistema final, integración y pruebas. Es esta fase, el sistema final de la aplicación tiene que ser validado y verificar que satisface todos los requerimientos especificados y todas las especificaciones añadidas durante el proceso de desarrollo.
Tiempo empleado = 168 horas

5. Mantenimiento, mejoras y soporte. Para poder evolucionar con el entorno, la aplicación necesita de esta fase, dado que el entorno de trabajo de una aplicación de Inteligencia Artificial es muy dinamica.
Tiempo empleado = indefinido.

El coste estimado de esta aplicación es ~~de unos~~ difícil de predecir ya que el valor en mercado de la AI esta in crechendo, ya que desde el 2017 al 2020 su valuación há subido un 70%, pero sería más o menos de unos 150.000,00 €