Tryton y el ciclo de vida del software

Nicolás López Solano

Indice

- ***Proyecto de implantación de ERP***
  - Implementación de requisitos
  - Más proyectos …
  - Mantenimiento
- ***Tryton***
  - Implementación
  - Gestión de la configuración
  - Requisitos

Datalife

Proyecto de implantación de ERP

- ***Complejo***
- ***Fases***
  - Consultoría
    - Importante → Requisitos
    - El ERP no es determinante
  - Implementación
  - Formación, arranque
  - Mantenimiento

Datalife

Implementación de requisitos

- ***Requisitos: Resultado de consultoría***
- ***Implementación***
  - ¿Lenguaje de programación?, ¿OOP? …
  - Reinvento la rueda o puedo usar el core del ERP
  - Me permite integrar fácil con hardware y software
  - Gestión de la configuración del software
- ***Aspectos: Requisitos transversales***

Mas proyectos ...

- ***Implementación de requisitos ...***
  - Mismo requisito en varios proyectos
  - Requisitos contradictorios
- ***Varios equipos de desarrollo***
  - En varios centro de trabajo
- ***¡¡Tengo un producto por cada proyecto!!***
  - Requisitos implementados formando parte de un único producto → ciclo de vida en espiral

Datalife

Mantenimiento

- ***Incorporación mejoras de nuevas versiones***
  - Compatible con los requisitos del cliente
  - Tiempo para preparar la migración
  - Tiempo de estabilización
- ***Incorporación de nuevas características***
  - ¿Entra a formar parte del core del ERP?

Tryton, implementación (I)

- ***Python***
  - OOP, propósito general, muy usado
  - Gran capacidad de integración, infinidad de librerías disponibles ...
- ***Amplia cobertura funcional***
  - Código fuente disponible
  - Procesos sencillos y robustos
    - Implementados en módulos (paquetes python)

Tryton, Implementación (II)

- *TDD*
  - Escenarios para implementar test correspondientes a procesos y requisitos

Gestión de la configuración

- *Cada módulo es un repositorio Mercurial o Git*
- *Fundación Tryton*
  - Infraestructura para el core de Tryton
    - hg.tryton.org
    - docs.tryton.org
    - disscuss.tryton.org
  - Varias equipos trabajando de forma distribuida y coordinada
  - Guía e inspiración para los integradores
    - ¡¡¡Si tu proceso se aleja demasiado, mal!!!

Tryton, Módulos

- ***Modularidad sin precedentes***
- ***Un módulo***
  - Es un paquete python
  - Implementa uno o varios requisitos
    - También sus test
  - Adapta o modifica el comportamiento de otros módulos
    - Procesos, vistas, traducciones, …
  - Incluso así, se puede hacer mal ...

Dependencias, Extras y Aspectos

- ***Dependencias***
  - Unos módulos dependen de otros
- ***Dependencias Extra***
  - Dependencia no obligatoria
  - El módulo implementa un comportamiento si la dependencia extra esta instalada
  - Imprescindible para modelar aspectos limpiamente
    - Ejemplo: Globalgap

Tryton, proyectos vs producto

- ***Mismo requisito en varios proyectos***
  - Implementar un módulo e implantar en varios proyectos
- ***Requisitos contradictorios***
  - Implementados en módulos distintos, que se implantarán en el proyecto adecuado
- ***Varios proyectos un único producto***
  - Ciclo de vida en espiral

Datalife

Tryton, nuevas versiones

- *Nueva versión cada 6 meses*
- *Los tests de cada módulo aseguran que este es compatible con la nueva versión*
- *Los tests garantizan que las nuevas implementaciones siguen respetando los requisitos previamente implementados*
- *Minimizan el tiempo de preparación de la migración. El cliente no percibe fallos*
- *Software de alta calidad*

Datalife

¿Alguna pregunta?

¡Muchas gracias!

www.datalifeit.es

info@datalifeit.es