

script.md

Glosario
- Paradigma: conjunto de conceptos/teorias/patrones/métodos y convenciones que pueden ser usados en un escenario específico

Presentacion
Buen día, mi nombre es Timoteo Ponce y tengo el agrado hoy de presentarles mi trabajo de grado para la maestría de Ing. en Software.

El trabajo se titula 'mejora de correctitud en el desarrollo de sistemas de software, aplicando el paradigma de desarrollo en roles DCI'

Problema
Empecemos por el problema, y como siempre es mejor iniciemos con un ejemplo.

Problema-1
Actualmente en este momento en el planeta tierra existen 300 millones de usuarios de MSWord,

Problema-2
y de estos usuarios aproximadamente 40 millones de documentos se crean y editan diariamente.

Problema-3
todos estos documentos tienen imágenes, localizadas y personalizadas, y entonces, a alguien se le ocurre cambiar el tamaño de página.

y ahora tenemos 40 millones de problemas, el documento pierde formato, los espacios se saltan, y las imágenes se reubican donde pueden.

Pero claro me dirán, esto se soluciona usando apropiadamente el software, pero he ahí un problema,

Problema-4
esos son uds. adaptandose a la herramienta, no la herramienta resolviendo un problema,

ud. verán, existe en el programa un conflicto, lo que el usuario quiere no está reflejado en el modelo de programación usado dentro del programa, el programa está descrito como una interpretación de lo que sus programadores entienden del usuario, y no realmente de lo que el espera.

Problema-5
qué tenemos entonces?
- inconsistencia entre lo que el usuario quiere y el programa hace
- necesidades constantes de soporte y mantenimiento
- costos elevados de soporte y mantenimiento, porque el programador debe actualizar el programa interpretando lo que el usuario espera
- costos de desarrollo con ingresos reducidos

A esto existe una teoría de Joel Spolsky que dice, 'todo software toma 10 años para ser un buen software'.

Solucion propuesta
Este es un gran problema, muchas empresas mueren antes de siquiera llegar a tener la chance de cumplir lo que el usuario final quiere, es por eso que la situación deseada de muchas empresas de software se puede resumir en:

Solucion propuesta-1
- programas consistentes con lo que los usuarios esperan (programas correctos)
- programas flexibles para evolucionar
- reducir costos de desarrollo

DCI
Ahora, entrando al trabajo en si, debemos dar un vistazo a DCI, qué es y cual es su valor para este trabajo.

Page 1

script.md

Page 2

DCI-1
La definición nos dice que:

"DCI es un esquema de diseño que a través de elementos naturales de comunicación, permite reflejar el comportamiento de una aplicación de software hacia los usuarios finales"

Como vemos en el gráfico, DCI es un esquema que permite tener los elementos que usamos al hablar, entidades, ambientes, etc. Como elementos de programación, es decir, DCI permite que plasmemos lo que el usuario quiere de forma directa en el modelo de programación.

DCI-2
Y cómo hace esto? bueno, en DCI un sistema se define por:
- lo que es: lo que es estable y no cambia
- y lo que hace: lo que es dinámico y cambia frecuentemente

Un poco confuso? tomemos un ejemplo

DCI-3
Aquí tenemos a una persona, llamemosla Daveyba, es un ser humano definido por su ADN, sus huellas digitales, su tipo de sangre, y muchas otras características que difícilmente cambian que la hacen única ante sus semejantes.

A estos elementos los denominamos DATOS.

DCI-4
Ahora, Daveyba va al trabajo todos los días, es una Administradora en su oficina, es una madre de familia en casa, es una conductora en carretera. Pero todas estas funciones no son lo que Daveyba es, estas son cosas que Daveyba hace!!

Estos son sus ROLES

DCI-5
Y todos estos roles tienen solamente un significado si se está en diferentes lugares. Sería interesante ver a Daveyba con un cuchillo en la oficina, no creen?...

Y a todos estos lugares donde un ROL tiene sentido, se los denomina CONTEXTOS.

DCI-6
Y la última palabra en DCI, INTERACCIONES, se definen por como Daveyba, en sus diferentes Roles, en sus respectivos Contextos, interactúa con otros ROLES!!

Menos Rory, en todos los contextos su esposa manda.

Entonces DCI nos dice que estos elementos de lenguaje natural nos permiten describir como una aplicación de software debe funcionar!!

DCI-7
Y para esto DCI recomienda el siguiente formato de especificación de lo que un usuario quiere:

"Daveyba quiere realizar su declaración de impuestos como trabajador dependiente durante el día, y como trabajador independiente cuando trabaja en casa por las noches"

Si a nosotros, como programadores de un sistema de declaración de impuestos nos entregan esta especificación de requerimiento, tenemos una descripción muy completa ahí!!

A este estilo de especificación se le denomina 'Narrativa de usuario', y es particularmente soportado por un artefacto muy conocido.

DCI-8
El caso de uso!!
A pesar de que los casos de uso ya no son tan populares, son herramientas excelentes para capturar narrativas de usuario, y

encaja perfectamente en lo que DCI propone, con todos sus elementos, roles y contextos.

DCI-9

Ok, ahora a lo práctico, cómo funciona DCI realmente dentro de un sistema de software, tomemos el ejemplo simple de un sistema bancario, tenemos dos casos de uso, transferir un depósito y cancelar una tarjeta.

Los círculos representan todos los objetos que toman parte en el proceso, y las flechas indican sus interacciones.

DCI-10

Ahora, separamos en áreas los posibles cursos de ejecución de estos dos casos de uso, como ven en la transferencia puede darse un solo caso, pero en la cancelación de tarjeta hay dos posibles alternativas de ejecución.

DCI-11

Entonces, DCI nos dice que saquemos estas áreas de ejecución y las convirtamos en contextos, que saquemos las operaciones de los objetos y los convirtamos en roles, y movamos las interacciones entre objetos hacia interacciones entre roles.

Dejando abajo los objetos como lo que son, solo datos que pueden asumir uno o varios Roles en diferentes contextos.

DCI-12

Si vemos el caso de Daveyba, Daveyba es una instancia de la entidad Persona, a la cual le acoplamos los roles de Contador, Auditor o Administrador para que cumpla una función dentro de un contexto.

En nuestro ejemplo 'elaborar un comprobante financiero diario', qué rol debería ser usado en este contexto?

DCI-13

Si vamos a mas bajo nivel, se vería así:

- Tenemos el contexto de comprobante financiero, cada caso de uso sería un contexto
- Obtenemos una persona y le acoplamos el rol de contador
- Procedemos a la ejecución del caso de uso

Si hacemos una analogía, se ve como el script de una película, donde existen actores y actrices, les decimos 'tú eres el villano y tú la heroína, estas son tus indicaciones' y decimos, 'listos, cámaras, acción!!'

DCI-14

Este es un modelo de programación simple en la superficie, pero para que funcione de esta manera se necesitan artefactos y abstracciones avanzadas que realicen el mapeo de los diferentes elementos de forma eficiente.

Y para lograr esto se tienen estas necesidades para un modelo de programación DCI:

- programación multi-paradigma (esencialmente orientada a objetos, funciona y orientada a aspectos)
- modelo computacional dinámico (que pueda ser cambiado sin detener el sistema de software)
- composición de objetos (ojo, no es herencia)
- contextos independientes de ejecución (o memoria transaccional)