

# Leyes de Lehman

# Introducción

- ▶ Mier Lehman y Laszlo Belady determinan unas leyes en los años 70-80
- ▶ El gasto estimado en programación en Estados Unidos en el año 1977 fue de unos 50-100 billones de dólares
- ▶ De este gasto el 30% fue en desarrollo de software y el 70% en el mantenimiento de este

# Introducción

- ▶ Llegaron a la conclusión de que el software tiene que estar cambiando y creciendo constantemente para poder adaptarse a los nuevos requisitos
- ▶ Al observar esto decidieron determinar unas leyes que estos cambios deberían seguir



# Clasificación programas

PARA COMPRENDER LA EVOLUCIÓN DEL SOFTWARE

# S-Program

- ▶ Especificación previa
- ▶ mcm, ocho reinas
- ▶ Objetivo: alcanzar solución deseada
- ▶ Secundario: eficiencia, elegancia...
- ▶ Estático debido a que no se puede alterar el mapeo entrada salida
- ▶ Corrección en función de la especificación

# P-Program

- ▶ Problema definido, pero solución aproximada al mundo real
- ▶ Ajedrez
- ▶ Corrección comparando con el entorno real
- ▶ Objetivo: conseguir una solución válida y que aporte valor en su contexto del mundo real

# E-Program

- ▶ Mecaniza actividad humana o social
- ▶ Incrustado en el mundo que modela
- ▶ Muy propenso a cambios (no especificación, retroalimentación, el programa afecta al mundo y viceversa)
- ▶ Sistemas operativos, comercio de acciones
- ▶ Por estos programas Lehman crea las siguientes leyes.

# Primera ley de Lehman. Cambio continuo

- ▶ «Los sistemas deben ser continuamente adaptados o se convierten progresivamente en menos satisfactorios».
- ▶ Adaptación al cambio
- ▶ No implica necesariamente crecimiento del sistema
- ▶ Factor económico y miedo a retocar un sistema que funciona.
- ▶ Mantenimiento inevitable



## Segunda ley de Lehman. Complejidad incremental

- ▶ «Cuando un sistema evoluciona se incrementa su complejidad a menos que se trabaje para mantenerla o reducirla».
- ▶ Complejidad crece a nivel de deuda técnica, administración, usabilidad y recursos software y hardware
- ▶ Buenas prácticas
- ▶ Coste mayor conforme se realicen nuevas actividades de mantenimiento

# Tercera ley de Lehman. Autorregulación

- ▶ «El proceso de evolución de un sistema es autorregulado con una distribución de las medidas del producto y del proceso cercana a la normal».
- ▶ Características del producto siguen una tendencia o son invariantes
- ▶ La complejidad interviene en el proceso autorregulador

# Cuarta ley de Lehman. Conservación de la estabilidad organizacional

- ▶ «La velocidad (y efectividad) de desarrollo de un sistema en evolución permanece invariante durante su ciclo de vida».
- ▶ Atributo que se autorregula
- ▶ Complicado que velocidad de desarrollo aumente

## Quinta ley de Lehman. Conservación de la familiaridad

- ▶ «Cuando un sistema evoluciona, todos aquellos que están asociados a él: desarrolladores, personal de ventas, usuarios, deben mantener un conocimiento de su contenido y comportamiento para tratar de conseguir que la evolución sea satisfactoria. Un crecimiento excesivo disminuye ese conocimiento. De ahí que el crecimiento incremental promedio permanezca invariante».
- ▶ El crecimiento del sistema se debe realizar de manera sostenida.

# Sexta ley de Lehman. Crecimiento continuo

- ▶ «Las funcionalidades del sistema tienen que crecer constantemente para mantener la satisfacción del usuario a lo largo de su ciclo de vida».
- ▶ Funcionalidades descartadas
- ▶ Cubrir problemas con nuevas funcionalidades
- ▶ Menos frecuente hoy en día

# Séptima ley de Lehman. Reducción de la calidad

- ▶ «La calidad de los sistemas comienza a disminuir a menos que se mantengan de forma rigurosa y se adapten a los cambios en su entorno de funcionamiento (operacional)».
- ▶ El usuario es cada vez más exigente
- ▶ Mayor complejidad del sistema
- ▶ Convivencia del sistema con su entorno (software y hardware)



# Octava ley de Lehman. Realimentación del sistema

- ▶ «El proceso de evolución del sistema es consecuencia de un proceso de realimentación (feedback) a diferentes niveles, de manera iterativa y por diferentes actores, y debe ser considerado como tal para conseguir mejoras significativas sobre cualquier base razonable».
- ▶ Con cada evolución del sistema se puede tener feedback para plantear diferentes acciones de mejora

FIN