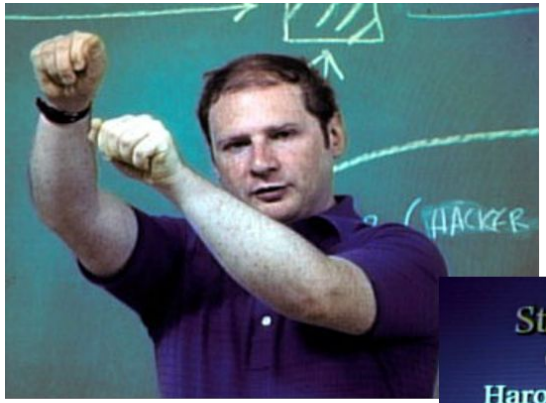
(**Conocimiento** expresado en un)

Modelo Computable
de un ***Dominio de Problema***
de la Realidad

¿Qué es el Desarrollo de Software?

¿Es una Ciencia?



http://groups.csail.mit.edu/mac/classes/6.001/abelson-sussman-lectures/

Comentarios

- No explica bien por qué no es una ciencia
- Dice que tiene que ver con magia…
  - Parece mágico pero no es magia
- Confusión entre herramientas y la esencia

¿Existe la ciencia de la computación?

Fred Brooks:

- Un dicho académico es que “Todo lo que se hace llamar ciencia, no lo es”.
  - “Física vs. Ciencias de la Computación”
- Ambos construimos, pero con distintos propósitos
  - El científico construye para aprender
  - El ingeniero aprende para construir

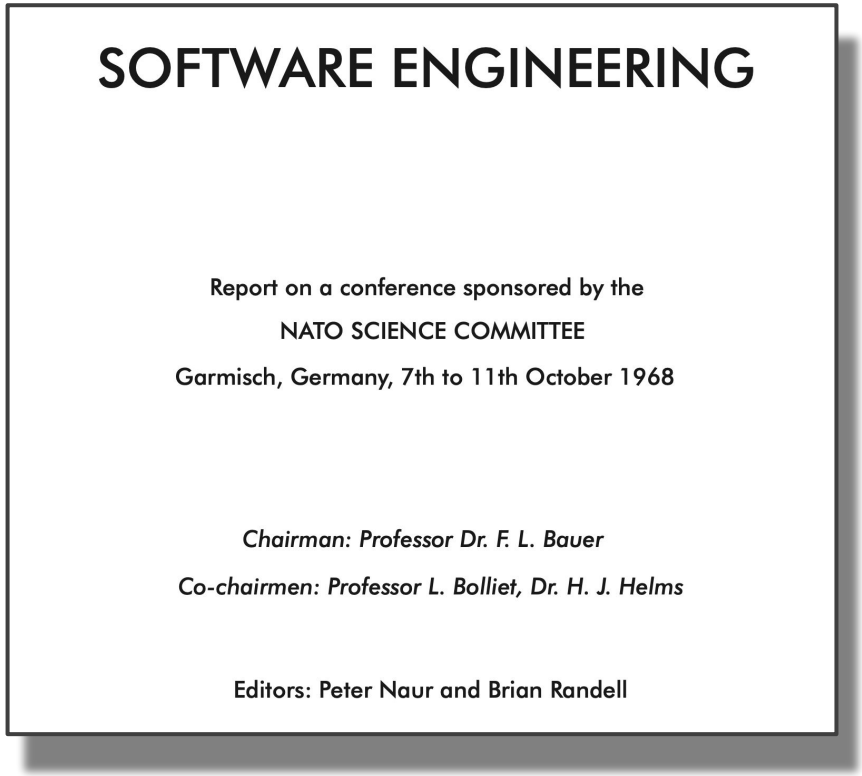
Fred Brooks. The computer scientist as a Toolsmith II. Communications of the ACM, Marzo 1996.

¿Es una Ingeniería?

Margaret Hamilton

Comencé a usar el término 'ingeniería de software' para distinguirlo del hardware y otros tipos de ingeniería…

¿Es una Ingeniería?



SOFTWARE ENGINEERING

Report on a conference sponsored by the

NATO SCIENCE COMMITTEE

Garmisch, Germany, 7th to 11th October 1968

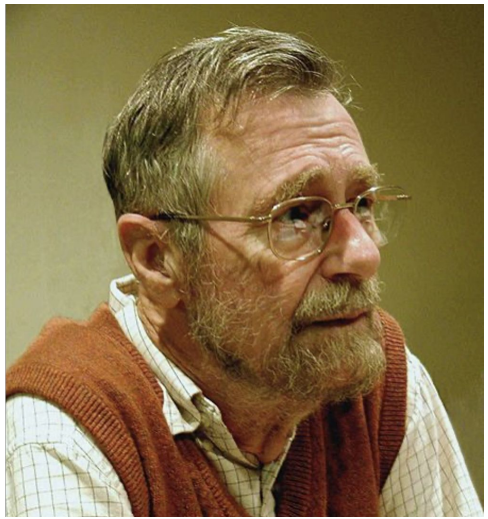
Chairman: Professor Dr. F. L. Bauer

Co-chairmen: Professor L. Bolliet, Dr. H. J. Helms

Editors: Peter Naur and Brian Randell

NATO, 1968, quería que lo fuese (Mary Shaw, IEEE Software1990)

¿Es una Ingeniería?



E. Dijkstra:

- Así como la economía se conoce como “La ciencia miserable”, la Ingeniería de Software debería ser conocida como “La disciplina condenada”.
- La ingeniería de software, por supuesto, se presenta a sí misma como otra causa meritoria, pero eso es puro cuento. Si uno lee cuidadosamente su literatura y analiza lo que sus devotos realmente hacen, descubre que la ingeniería de software ha aceptado como su carta de presentación “como programar si no puede”.
- Es realmente útil ver un programa como una fórmula. Primero, pone la tarea del programador en perspectiva: tiene que derivar esa fórmula... manipulación de símbolos usualmente llamada 'programación'.

Fuente: ver http://www.cs.utexas.edu/users/EWD/transcriptions/EWD13xx/EWD1305.html

¿Es una Ingeniería?

Ingeniería de Software según Yourdon:

- La construcción de sistemas de software que funcionen no es una “ciencia”. Los así llamados “científicos en computación” nos tratan de convencer de que nuestros sistemas son en realidad grafos dirigidos, o n-tuplas de formas normalizadas, o autómatas de estados finitos... Sus pronunciamientos son más relevantes a un Zen que al difícil problema de construir sistemas y programas útiles y fáciles de mantener.
- En lugar de aproximaciones académicas, lo que necesitamos es un conjunto de métodos prácticos que traten con la naturaleza propensa a errores del personal de proyectos, los usuarios, los encargados del mantenimiento, y el proceso de desarrollo. Ese conjunto de métodos es llamado “Ingeniería de Software”.

Fuente: ver http://www.yourdon.com

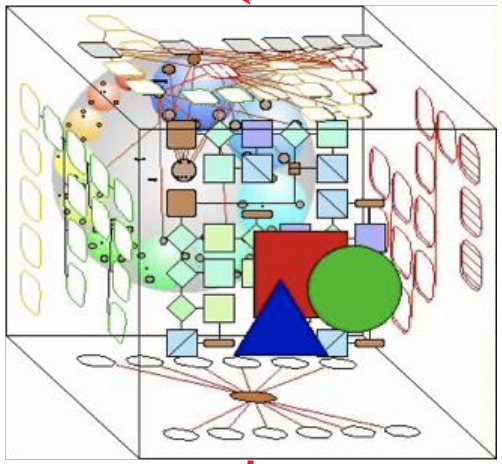
“En ciencia, si sabes lo que estás haciendo, no deberías estar haciéndolo. En ingeniería, si no sabes lo que estás haciendo, no deberías estar haciéndolo”

The Art of Science and Engineering
Richard Hamming

Entonces…
¿Qué es el Desarrollo de Software?

Desarrollo de Software

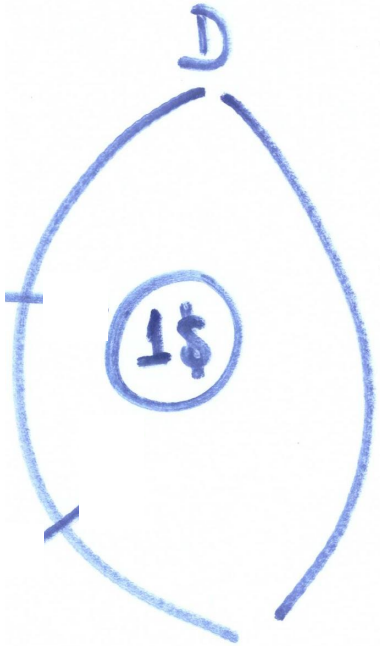
Algo único y particular

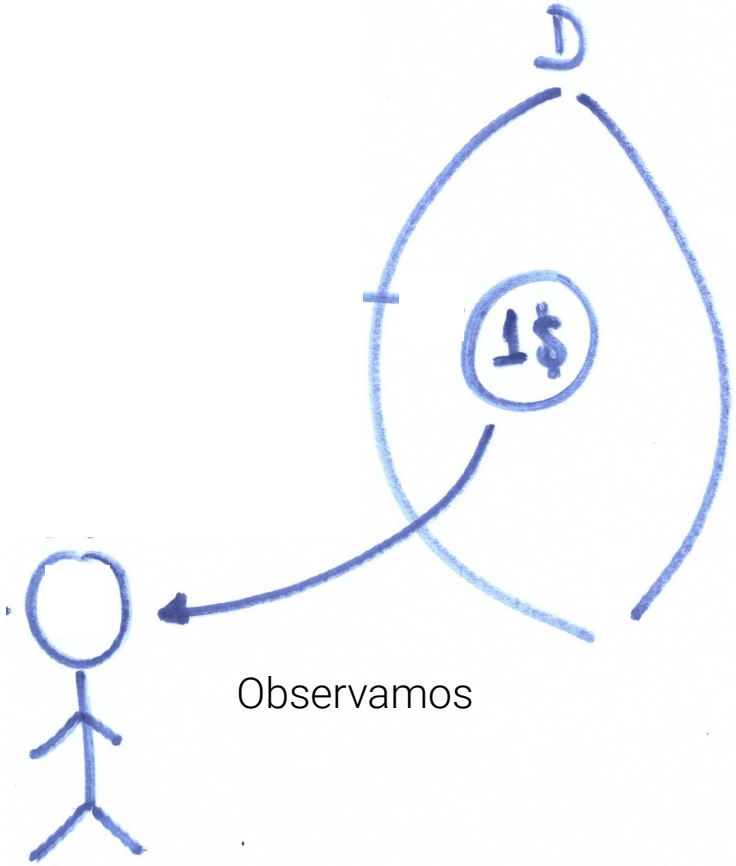


Mejor analogía: Proceso de Aprendizaje

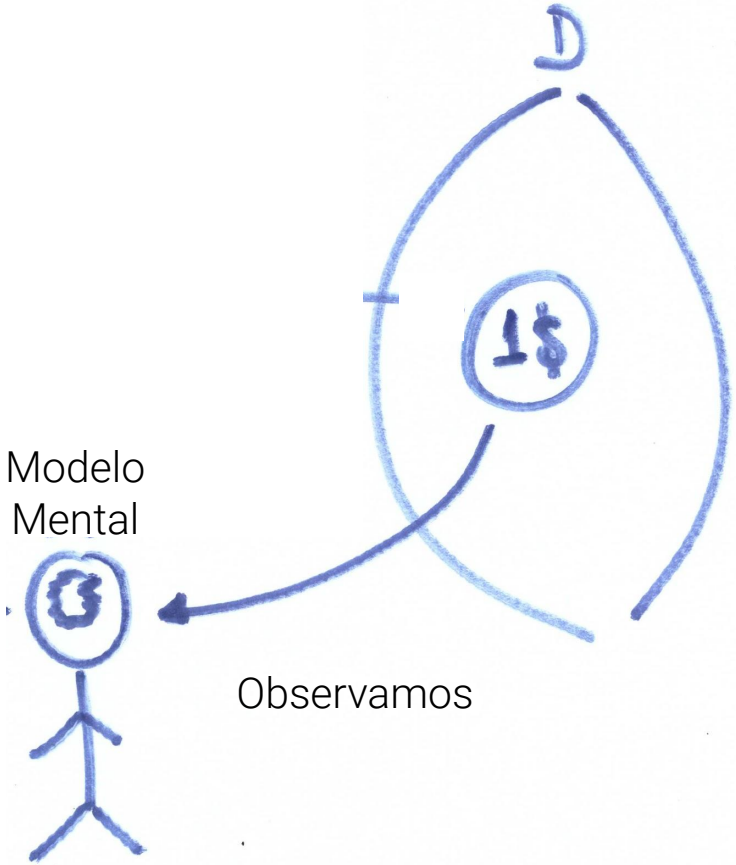
¿Cómo “creamos” un modelo?

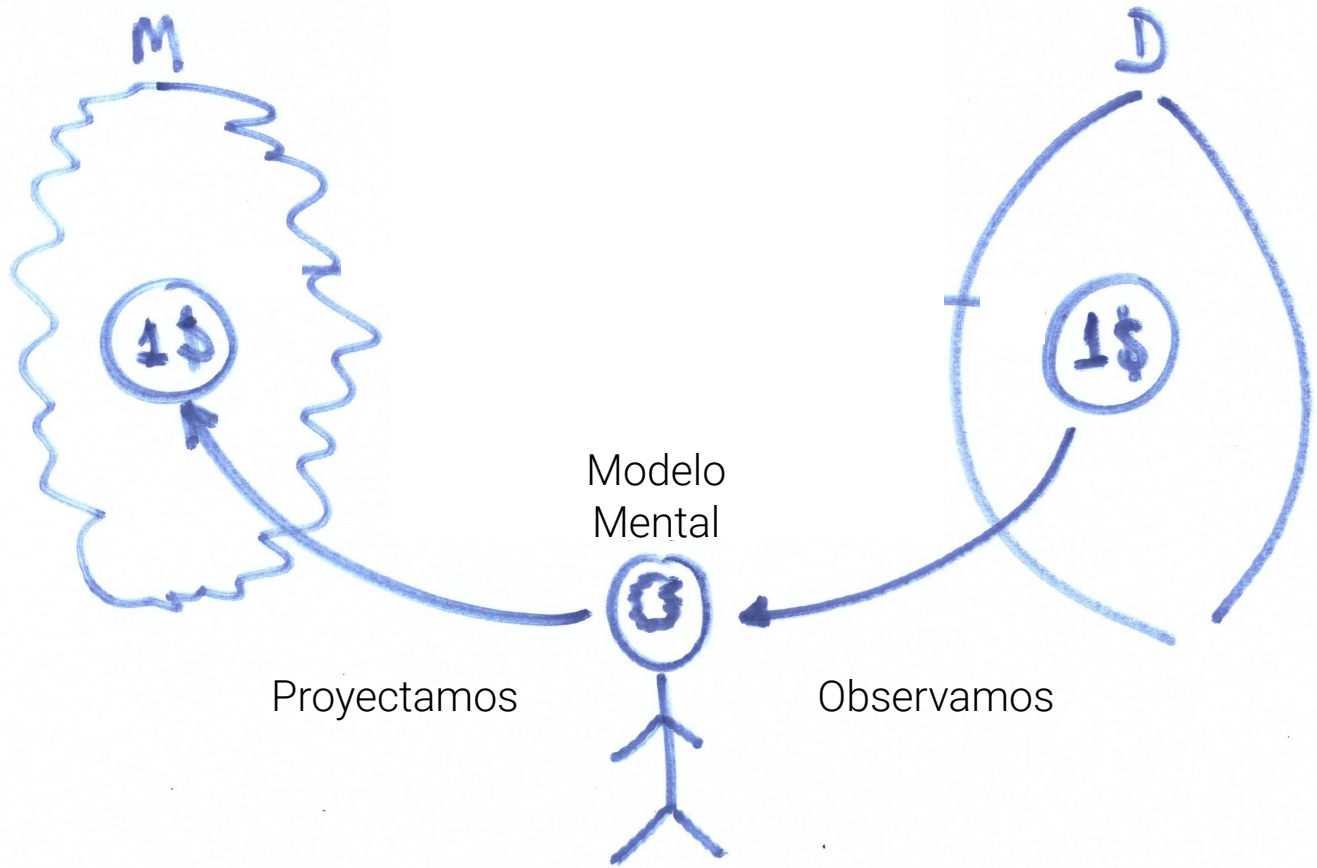
¿Cómo Desarrollamos Software?

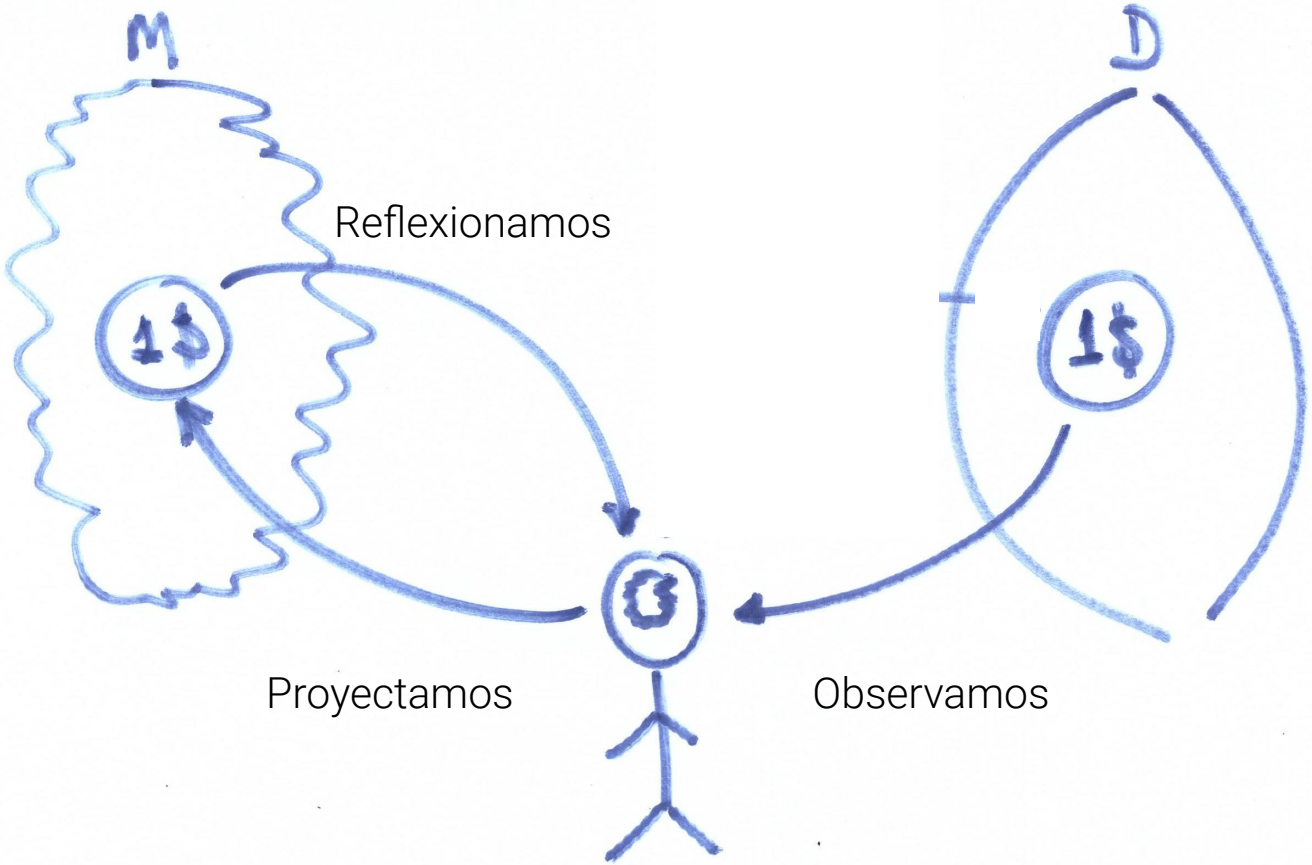

D
1$

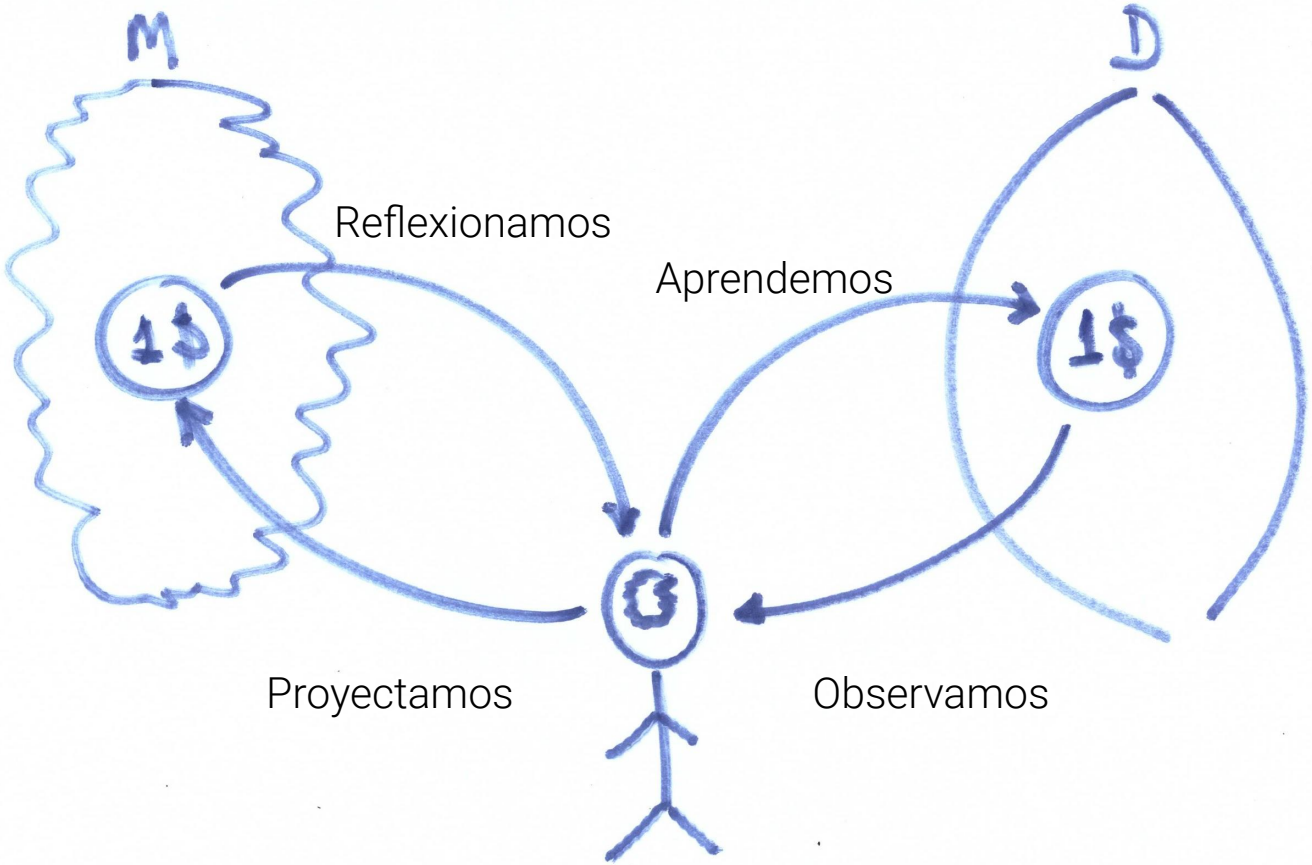


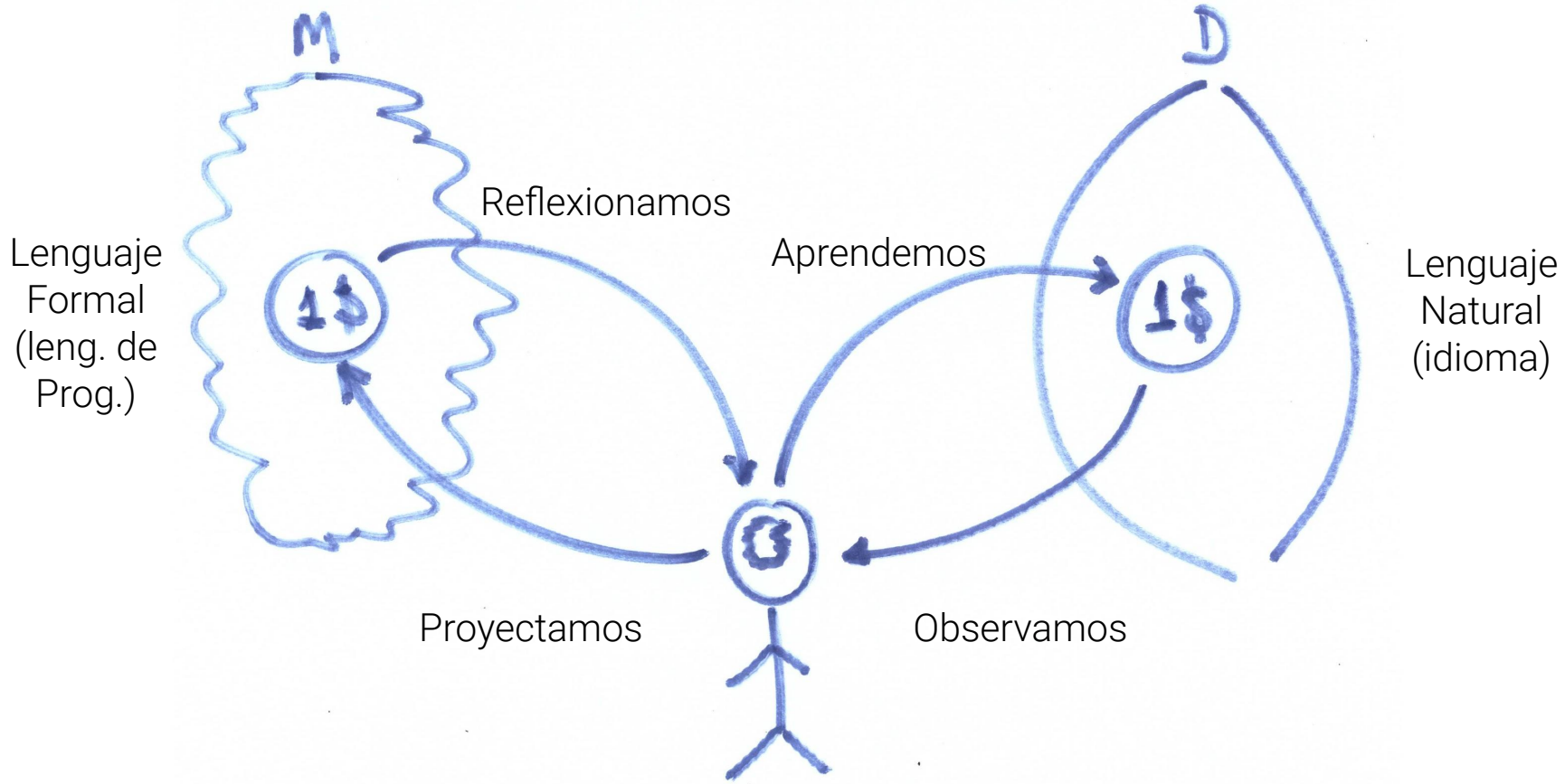
Observamos


1$
Modelo
Mental
Observamos

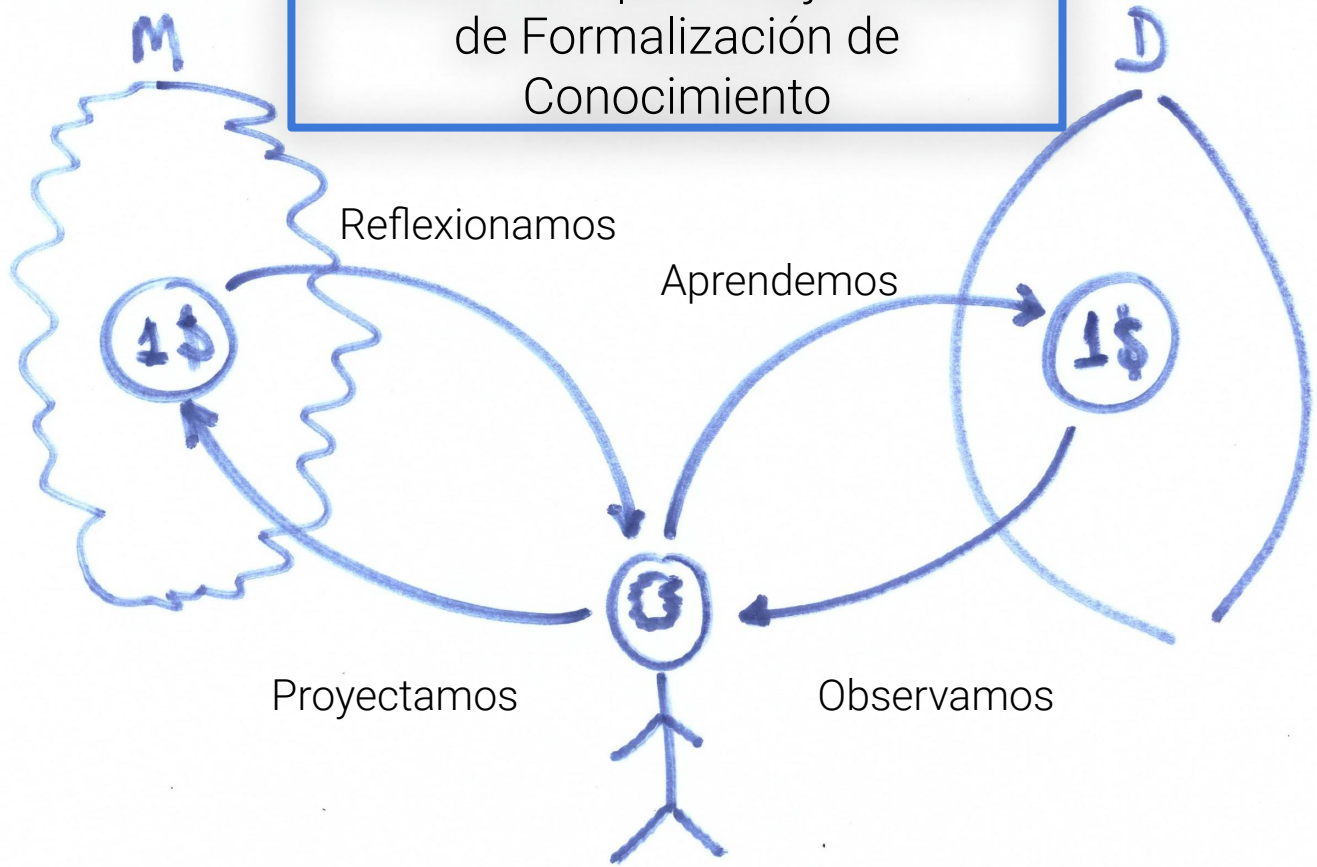

M
D
1$
1$
Modelo
Mental
Proyectamos
Observamos


M
D
Reflexionamos
1$
1$
Proyectamos
Observamos


M
D
Reflexionamos
Aprendemos
1$
1$
Proyectamos
Observamos


M
D
Reflexionamos
Aprendemos
Lenguaje Formal (leng. de Prog.)
Lenguaje Natural (idioma)
Proyectamos
Observamos

Proceso de Aprendizaje a través de Formalización de Conocimiento

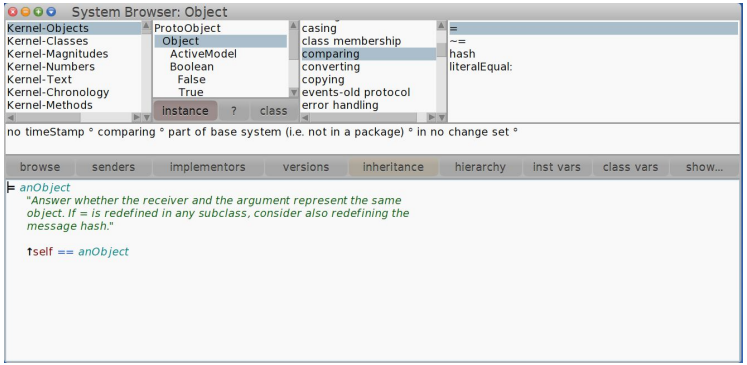


Característica fundamental: Paso del Tiempo en el Desarrollo

Paso del Tiempo y Aprendizaje

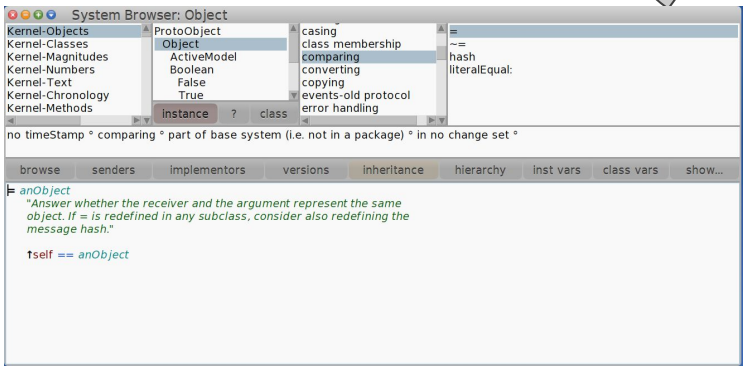
Para Entender

Diseñamos



t 0 . . . . . t n

Transmisión de Conocimiento



Es **fundamental** que nuestros modelos “**nos enseñen**”

Desarrollo de Software

El desarrollo de software es un proceso de aprendizaje, lo que implica:

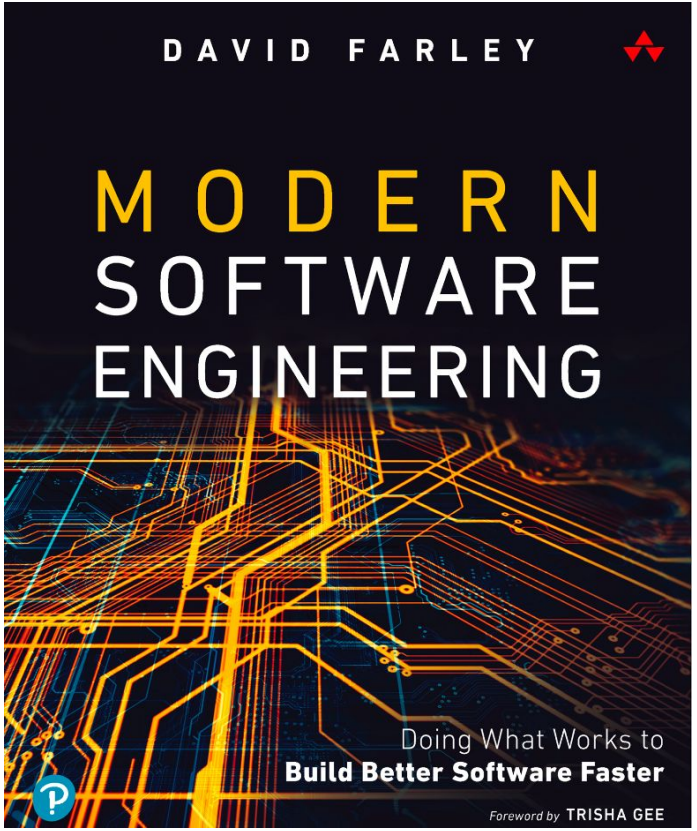
- Es iterativo
- Es incremental
- El conocimiento se genera a partir de hechos concretos
- El conocimiento generado debe ser organizado

Desarrollo de Software

Características

- El dominio de problema está generalmente especificado en lenguajes ambiguos y contextuales (ej. Lenguaje natural)
- El proceso de desarrollo implica **desambiguar** y **descontextualizar** el conocimiento del dominio de problema
- El proceso de desarrollo implica hacer **explícito** y **externo** el conocimiento implícito e internalizado de los expertos de dominio
- El **CAMBIO** es una característica esencial del software, no accidental porque:
  - Cambia el dominio de problema
  - Cambia nuestro entendimiento del dominio de problema
  - Cambia la manera de modelar lo que entendemos del dominio de problema

Desarrollo de Software



Desarrollo de Software

Ingeniería – La aplicación práctica de la Ciencia

El desarrollo de software es un proceso de descubrimiento y exploración; por lo tanto, para tener éxito en ello, los ingenieros de software deben convertirse en expertos en el aprendizaje.

El mejor enfoque de la humanidad para el aprendizaje es la ciencia, por lo que debemos adoptar las técnicas y estrategias de la ciencia y aplicarlas a nuestros problemas. Esto a menudo se malinterpreta en el sentido de que debemos convertirnos en físicos midiendo cosas a niveles de precisión irrazonables, en el contexto del software. La ingeniería es más pragmática que eso.

Lo que quiero decir cuando digo que debemos aplicar las técnicas y estrategias de la ciencia es que debemos aplicar algunas ideas bastante básicas, pero sin embargo extremadamente importantes.

Wikipedia describe el método científico que la mayoría de nosotros aprendimos en la escuela como:

- **Caracterizar**: Hacer una observación del estado actual.
- **Formular una hipótesis**: Cree una descripción, una teoría que pueda explicar su observación.
- **Predecir**: Haz una predicción basada en tu hipótesis.
- **Experimentar**: Prueba tu predicción.

Desarrollo de Software

Ingeniería – La aplicación práctica de la Ciencia

El desarrollo de software es un proceso de descubrimiento y exploración; por lo tanto, para tener éxito en ello, los ingenieros de software deben convertirse en expertos en el aprendizaje.

El mejor enfoque de la humanidad para el aprendizaje es la ciencia, por lo que debemos adoptar las técnicas y estrategias de la ciencia y aplicarlas a nuestros problemas. Esto a menudo se malinterpreta en el sentido de que debemos convertirnos en físicos midiendo cosas a niveles de precisión irrazonables, en el contexto del software. La ingeniería es más pragmática que eso.

Lo que quiero decir cuando digo que debemos aplicar las técnicas y estrategias de la ciencia es que debemos aplicar algunas ideas bastante básicas, pero sin embargo extremadamente importantes.

Wikipedia describe el método científico que la mayoría de nosotros aprendimos en la escuela como:

- **Caracterizar**: Hacer una observación del estado actual.
- **Formular una hipótesis**: Cree una descripción, una teoría que pueda explicar su observación.
- **Predecir**: Haz una predicción basada en tu hipótesis.
- **Experimentar**: Prueba tu predicción.

Desarrollo de Software

Ingeniería – La aplicación práctica de la Ciencia

El desarrollo de software es un proceso de descubrimiento y exploración; por lo tanto, para tener éxito en ello, los ingenieros de software deben convertirse en expertos en el aprendizaje.

El mejor enfoque de la humanidad para el aprendizaje es la ciencia, por lo que debemos adoptar las técnicas y estrategias de la ciencia y aplicarlas a nuestros problemas. Esto a menudo se malinterpreta en el sentido de que debemos convertirnos en físicos midiendo cosas a niveles de precisión irrazonables, en el contexto del software. La ingeniería es más pragmática que eso.

Lo que quiero decir cuando digo que debemos aplicar las técnicas y estrategias de la ciencia es que debemos aplicar algunas ideas bastante básicas, pero sin embargo extremadamente importantes.

Wikipedia describe el método científico que la mayoría de nosotros aprendimos en la escuela como:

- **Caracterizar**: Hacer una observación del estado actual.
- **Formular una hipótesis**: Cree una descripción, una teoría que pueda explicar su observación.
- **Predecir**: Haz una predicción basada en tu hipótesis.
- **Experimentar**: Prueba tu predicción.

Desarrollo de Software

La ingeniería de software *es la aplicación de un enfoque científico empírico para encontrar soluciones económicas y eficientes a problemas prácticos de software.*

La adopción de un enfoque de ingeniería para el desarrollo de software es importante por dos razones principales. Primero, el desarrollo de software es siempre un ejercicio de descubrimiento y aprendizaje, y Segundo si nuestro objetivo es ser "eficientes" y "económicos", entonces nuestra capacidad de aprender debe ser sostenible.
Esto significa que debemos gestionar la complejidad de los sistemas que creamos de forma que mantengamos nuestra capacidad de aprender cosas nuevas y adaptarnos a ellas.
Entonces, debemos convertirnos en expertos en el aprendizaje y expertos en el manejo de la complejidad.
Hay cinco técnicas que forman las raíces de este enfoque en el aprendizaje. En concreto, para convertirse expertos en aprendizaje, necesitamos lo siguiente:

- Iteración
- Comentario
- Incrementalismo
- Experimentación
- Empirismo

Feedback Inmediato

Inventing on principle: http://vimeo.com/36579366

Comentarios

- Jugar a ser computadoras
- Generar una intuición
- Encontrar errores más rápidamente

Feedback inmediato = Fundamental en el proceso de aprendizaje

Conclusiones

Software = Modelo Computable

Desarrollo de Software = Proceso de Aprendizaje