

Cómo tener una Mala Carrera Profesional en Investigación/ Academia

Profesor David A. Patterson

Febrero 2002

www.cs.berkeley.edu/~pattrsn/talks/nontech.html

Resumen

- **Parte I: Consejos Clave para una Mala Carrera Académica**
- **Parte II: Consejos Clave en Alternativas para una Mala Carrera Académica**
- **Tópicos cubiertos en Parte II**
 - **Seleccionando un Problema**
 - **Escogiendo una Solución**
 - **Realizar la Investigación**
 - **Evaluando los Resultados**
 - **Comunicando los Resultados**
 - **Transfiriendo Tecnología**

Jugada #1 para una Mala Carrera Académica: Sé EL experto líder

- **¡Inventa un nuevo campo!**
 - **Asegúrate que sea solo un poco diferente**
- **Sé el verdadero Llanero Solitario: No trabajes con otros**
 - **No hay ambigüedad en los créditos**
 - **Adopta la personalidad de Prima Donna**
- **Horizontes de Investigación**
 - **Nunca defines un criterio de éxito**
 - **Evita resultados finales en menos de 20 años**
 - **Apégate a un solo tópico durante toda tu carrera**
 - **Aun si la tecnología parece que dejarte atrás, mantente con tu mismo problema**

Anunciando un campo nuevo en Sistemas Operativos: “**Sistemas basados en discapacidades**”

- **La seguridad computacional es cada vez más importante**
 - **Intuición: resolver en base a capacidades es casi correcto**
- **Idea: ¡Crear listas de cosas que un proceso NO puede hacer!**
- **Pregunta Clave:**
¿Debería almacenar las discapacidades con cada usuario o con los objetos que no pueden acceder?
- **Otros tópicos: discapacidades encriptadas, resolución en base a discapacidades**
- **Comience una nueva secuencia de cursos y una nueva revista sobre Teoría y Práctica de Sistemas basados en discapacidades**

Jugada #2 para una Mala Carrera Académica: Déjate guiar por la Complejidad (Confunde a tus Enemigos)

- **El mejor cumplido:
“Es tan complicado, que no puedo entender tus ideas”**
- **Es más fácil reclamar crédito por buenas ideas
subsecuentes**
 - **Si nadie te entiende, ¿cómo pueden contradecir tus
propuestas?**
- **Es más fácil ser complicado**
- **Si no fuera complicado entonces ¿cómo podrían los
distinguidos colegas en departamentos alrededor del mundo
positivamente apreciar tanto tu extraordinaria capacidad
intelectual sobre pequeños detalles, como la profundidad de
tu contribución?**

Jugada #3 para una Mala Carrera Académica: Que nunca demuestren que estás equivocado

- **Evita implementaciones**
- **Evita Experimentos Cuantitativos**
 - Si tienes una buena intuición, ¿quién necesita experimentos?
 - ¿Porqué alimentar el “molino de la crítica”?
 - Toma mucho tiempo hacer mediciones
- **Evita Benchmarks**
- **Proyectos cuyos resultados finales sean ≥ 20 años te dan 19 años de seguridad**

Jugada #4 para una Mala Carrera Académica: Usa el Método Científico de la Computación

El Obsoleto Método Científico

- Hipótesis
- Secuencia de experimentos
- Cambiar 1 parámetro/exp.
- Aprobar/Desaprobar Hipótesis
- Documentar para que otros reproduzcan los resultados

El Método Científico de la Computación

- Corazonada
- Haz 1 experimento y cambia todos los parámetros
- Deséchalo si no soportan la corazonada
- ¿Porqué perder el tiempo? Esto ya se conoce

Jugada #5 para una Mala Carrera Académica: No te distraigas por otros (Evita realimentación)

- **Siempre domina las conversaciones: el silencio es ignorancia**
 - **Corolario: Ser gritón es ser muy listo**
- **No leas**
- **No estés tentado en interactuar con usuarios, o con la industria**
- **Revisiones**
 - **Si es simple y obvio en retrospectiva => Recházalo**
 - **Los resultados cuantitativos no importan si no te muestran lo que ya sabes => Recházalos**
 - **Todo lo demás => Recházalo**

Jugada #6 para una Mala Carrera Académica: Publicar Artículos de Revista ES Transferencia de Tecnología

- **Ten como objetivo las revistas con alto impacto: son la Moneda de Cambio en el Reino Académico**
 - Toma de 2 a 3 años ir desde someter hasta publicar => atemporal
- **Como el científico líder, tu trabajo es publicar en revistas; no es tu trabajo hacer tus ideas accesibles al ingeniero ordinario**
- **Ir a conferencias y visitar compañías es nada más perder tiempo valioso de investigación**
 - Tiempo de viaje, hacer interacción con otros, servir en comités de programa, ...

Jugada #7 para una Mala Carrera Académica: Tácticas de escritura

- **Artículos: Es la cantidad, no la calidad**
 - **Éxito personal = Longitud de la lista de publicaciones**
 - **“La LPU (Least Publishable Unit: Unidad Mínima Publicable) es buena para tí”**



- **La productividad de un estudiante = número de artículos que produce**
 - **Número de estudiantes: mientras más, mejor**
 - **Nunca pida implementar a los estudiantes: reduce el número de artículos**

- **Legalmente cambia tu nombre a Aaaanderson**

Los Cinco Mandamientos de la Escritura para una Mala Carrera Académica

- I. No definirás términos, ni explicarás nada.**
- II. No reemplazarás “vamos a hacer” con “hemos hecho”.**
- III. No mencionarás las desventajas de tu aproximación.**
- IV. No harás referencia a otros artículos.**
- V. Publicarás antes que implementar nada.**

Los Siete Mandamientos de Presentación para una Mala Carrera Académica

- I. No ilustrarás.
- II. No codiciarás la brevedad.
- III. No imprimirás en letra grande.
- IV. No usarás **color**.
- V. Cubrirás tus diapositivas desnudas.
- VI. No saltarás diapositivas en una presentación larga.
- VII. No practicarás.

Siguiendo los mandamientos

- We describe the philosophy and design of the control flow machine, and present the results of detailed simulations of the performance of a single processing element. Each factor is compared with the measured performance of an advanced von Neumann computer running equivalent code. It is shown that the control flow processor compares favorably in the program.
- We present a denotational semantics for a logic program to construct a control flow for the logic program. The control flow is defined as an algebraic manipulator of idempotent substitutions and it virtually reflects the resolution deductions. We also present a bottom-up compilation of medium grain clusters from a fine grain control flow graph. We compare the basic block and the dependence sets algorithms that partition control flow graphs into clusters.
- A hierarchical macro-control-flow computation allows them to exploit the coarse grain parallelism inside a macrotask, such as a subroutine or a loop, hierarchically. We use a hierarchical definition of macrotasks, a parallelism extraction scheme among macrotasks defined inside an upper layer macrotask, and a scheduling scheme which assigns hierarchical macrotasks on hierarchical clusters.
- We apply a parallel simulation scheme to a real problem: the simulation of a control flow architecture, and we compare the performance of this simulator with that of a sequential one. Moreover, we investigate the effect of modeling the application on the performance of the simulator. Our study indicates that parallel simulation can reduce the execution time significantly if appropriate modeling is used.
- We have demonstrated that to achieve the best execution time for a control flow program, the number of nodes within the system and the type of mapping scheme used are particularly important. In addition, we observe that a large number of subsystem nodes allows more actors to be fired concurrently, but the communication overhead in passing control tokens to their destination nodes causes the overall execution time to increase substantially.
- The relationship between the mapping scheme employed and locality effect in a program are discussed. The mapping scheme employed has to exhibit a strong locality effect in order to allow efficient execution
- Medium grain execution can benefit from a higher output bandwidth of a processor and finally, a simple superscalar processor with an issue rate of ten is sufficient to exploit the internal parallelism of a cluster. Although the technique does not exhaustively detect all possible errors, it detects nontrivial errors with a worst-case complexity quadratic to the system size. It can be automated and applied to systems with arbitrary loops and nondeterminism.

Los Siete Mandamientos de Posters para una Mala Carrera Académica

- I. No ilustrarás.
- II. No codiciarás la brevedad.
- III. No imprimirás en letra grande.
- IV. No usarás color.
- V. No atraerás atención a tí mismo.
- VI. No prepararás una corta presentación.
- VII. No te prepararás con antelación.

Siguiendo todos los mandamientos

How to Do a Bad Poster
David Patterson
University of California
Berkeley, CA 94720

We describe the philosophy and design of the control flow machine, and present the results of detailed simulations of the performance of a single processing element. Each factor is compared with the measured performance of an advanced von Neumann computer running equivalent code. It is shown that the control flow processor compares favorably in the program.

We present a denotational semantics for a logic program to construct a control flow for the logic program. The control flow is defined as an algebraic manipulator of idempotent substitutions and it virtually reflects the resolution deductions. We also present a bottom-up compilation of medium grain clusters from a fine grain control flow graph. We compare the basic block and the dependence sets algorithms that partition control flow graphs into clusters.

Our compiling strategy is to exploit coarse-grain parallelism at function application level: and the function application level parallelism is implemented by fork-join mechanism. The compiler translates source programs into control flow graphs based on analyzing flow of control, and then serializes instructions within graphs according to flow arcs such that function applications, which have no control dependency, are executed in parallel.

A hierarchical macro-control-flow computation allows them to exploit the coarse grain parallelism inside a macrotask, such as a subroutine or a loop, hierarchically. We use a hierarchical definition of macrotasks, a parallelism extraction scheme among macrotasks defined inside an upper layer macrotask, and a scheduling scheme which assigns hierarchical macrotasks on hierarchical clusters.

We apply a parallel simulation scheme to a real problem: the simulation of a control flow architecture, and we compare the performance of this simulator with that of a sequential one. Moreover, we investigate the effect of modeling the application on the performance of the simulator. Our study indicates that parallel simulation can reduce the execution time significantly if appropriate modeling is used.

We have demonstrated that to achieve the best execution time for a control flow program, the number of nodes within the system and the type of mapping scheme used are particularly important. In addition, we observe that a large number of subsystem nodes allows more actors to be fired concurrently, but the communication overhead in passing control tokens to their destination nodes causes the overall execution time to increase substantially.

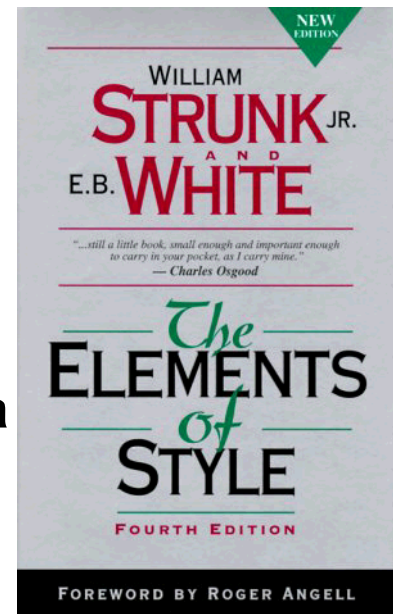
The relationship between the mapping scheme employed and locality effect in a program are discussed. The mapping scheme employed has to exhibit a strong locality effect in order to allow efficient execution. We assess the average number of instructions in a cluster and the reduction in matching operations compared with fine grain control flow execution.

Medium grain execution can benefit from a higher output bandwidth of a processor and finally, a simple superscalar processor with an issue rate of ten is sufficient to exploit the internal parallelism of a cluster. Although the technique does not exhaustively detect all possible errors, it detects nontrivial errors with a worst-case complexity quadratic to the system size. It can be automated and applied to systems with arbitrary loops and nondeterminism.

Resumen

- **Parte I: Consejos Clave para una Mala Carrera Académica**
- **Parte II: Consejos Clave sobre Alternativas para una Mala Carrera Académica**
- **Tópicos cubiertos en Parte II**
 - **Seleccionando un Problema**
 - **Escogiendo una Solución**
 - **Realizar la Investigación**
 - **Evaluando los Resultados**
 - **Comunicando los Resultados**
 - **Transfiriendo Tecnología**

Alternativas para Escribir Artículos



- **Haz lo opuesto a los mandamientos para la escritura**
Define términos, distingue entre “vamos a hacer” vs “hemos hecho”, menciona desventajas, desempeño real, haz referencia a otros artículos.
- **Encuentra trabajo relacionado con la búsqueda en línea INSPEC en lugar de solo en la red**
www.dbs.cdlib.org
- **Lee primero a Strunk and White, y sigue los pasos;**
 1. Haz un resumen de 1 página del artículo, con una sección tentativa de presupuesto/sección
 2. Haz un mapa de los párrafos
 - » 1 frase/enunciado por párrafo, figuras a mano con leyendas
 3. (Re)escribe el borrador
 - » Las leyendas/figura largas pueden contener detalles ~ Scientific American
 - » Usa Tablas para presentar datos que puedan hacer árido el texto
 4. Lee en voz alta, usa el corrector de ortografía y de gramática (MS Word; Bajo Herramientas, selecciona Gramática, selecciona Opciones, selecciona “técnico” para estilo de escritura vs. “estándar”; selecciona Opciones y seleccionar)
 5. Obtén realimentación y crítica sobre el borrador de amigos; regresa a 3.
- www.cs.berkeley.edu/~pattrsn/talks/writingtips.html

Alternativas para Presentaciones

- **Haz lo opuesto a los mandamientos**
 - I. No ilustrarás.
 - II. No codiciarás la brevedad.
 - III. No imprimirás en letra grande.
 - IV. No usarás color.
 - V. Cubrirás tus diapositivas desnudas.
 - VI. No saltarás diapositivas en una presentación larga.
 - VII. No practicarás.
- **Considera 2 minutos por diapositiva, deja tiempo para preguntas**
- **No sobre-animes**
- **Haz pruebas con amigos/críticos para obtener realimentación,**
 - incluye preguntas de una audiencia difícil
- **Graba una presentación de práctica (audio o video)**
 - » No memorices el discurso, pero ten notas listas
- **Bill Tetzlaff, IBM: “Dar una ‘presentación profesional’ de primera clase es la parte más importante de un viaje para entrevista. Tener a alguien que sabe que puede dar una presentación excelente de antemano incrementa grandemente las posibilidades de una invitación. Eso significa grandes presentaciones en conferencias.”**

Alternativas para Posters (de Randy Katz)

- **Responde las Cinco Preguntas de Heilmeier**
 1. ¿Cuál es el problema al que te enfrentas?
 2. ¿Cuál es el estado del arte actualmente?
 3. ¿Cuál es tu concepto o tecnología clave que “hace la diferencia”?
 4. ¿Qué has logrado hasta ahora?
 5. ¿Cuál es tu plan para lograr el éxito?
- **Haz lo opuesto a los mandamientos de Posters**
 - Un Poster intenta llamar la atención de la gente que pasa
- **Un poster de 9 páginas se ve como sigue:**

Enunciado del Problema	Estado del Arte	Concepto Clave
Logro # 1	Título y Logo visual	Logro # 2
Logro # 3	Plan para el éxito	Resumen y Conclusión

ROC: Recovery-Oriented Computing

Aaron Brown and David Patterson

ROC Research Group, EECS Division, University of California at Berkeley

For more info: <http://roc.cs.berkeley.edu>

AME is the 21st Century Challenge

• Availability

- systems should continue to meet quality of service goals despite hardware and software failures

• Maintainability

- systems should require only minimal ongoing human administration, regardless of scale or complexity: Today, cost of maintenance = 10X cost of purchase

• Evolutionary Growth

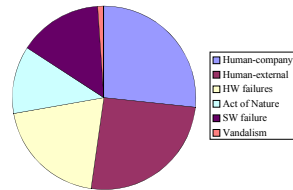
- systems should evolve gracefully in terms of performance, maintainability, and availability as they are grown/upgraded/expanded

• Performance was the 20th Century Challenge

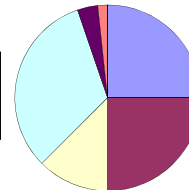
- 1000X Speedup suggests problems are elsewhere

People are the biggest challenge

Number of Outages



Minutes of Failure



• People > 50% outages/minutes of failure

- "Sources of Failure in the Public Switched Telephone Network," Kuhn; IEEE Computer, 30:4 (Apr 97)
- FCC Records 1992-1994; Overload (not sufficient switching to lower costs) + 6% outages, 44% minutes

Recovery-Oriented Computing (ROC) Hypothesis

"If a problem has no solution, it may not be a problem, but a fact, not to be solved, but to be coped with over time"
— Shimon Peres

• Failures are a fact, and recovery/repair is how we cope with them

• Improving recovery/repair improves availability

$$\text{Availability} = \frac{\text{MTTF}}{(\text{MTTF} + \text{MTTR})}$$

- Since $\text{MTTF} \gg \text{MTTR}$, 1/10th MTTR just as valuable as 10X MTBF

• Since major Sys Admin job is recovery after failure, ROC also helps with maintenance

ROC Principles: (1) Isolation and redundancy

• System is partitionable

- to isolate faults
- to enable online repair/recovery
- to enable online HW growth/SW upgrade
- to enable operator training/expand experience on portions of real system
- Techniques: Geographically replicated sites, Shared-nothing cluster, Separate address spaces inside CPU

• System is redundant

- sufficient HW redundancy/data replication => part of system down but satisfactory service still available
- enough to survive 2nd failure or more during recovery
- Techniques: RAID-6; N-copies of data

ROC Principles: (2) Online verification

• System enables input insertion, output check of all modules (including fault insertion)

- to check module operation to find failures faster
- to test correctness of recovery mechanisms
 - » insert faults and known-incorrect inputs
 - » also enables availability benchmarks
- to test if proposed solution fixed the problem
 - » discover whether need to try another solution
- to discover if warning systems are broken
- to expose and remove latent errors from each system
- to train/expand experience of operator
- Techniques: Global invariants; Topology discovery; Program checking (SW ECC)

ROC Principles: (3) Undo Support

• ROC system should offer Undo

- to recover from operator errors
 - » undo is ubiquitous in productivity apps
 - » should have "undo for maintenance"
- to recover from inevitable SW errors
 - » restore entire system state to pre-error version
- to recover from operator training via fault-insertion
- to replace traditional backup and restore
- Techniques: Checkpointing; Logging; and time travel (log structured) file systems

ROC Principles: (4) Diagnosis Support

• System assists human in diagnosing problems

- root-cause analysis to suggest possible failure points
 - » track resource dependencies of all requests
 - » correlate symptomatic requests with component dependency model to isolate culprit components
- "health" reporting to detect failed/failing components
 - » failure information, self-test results propagated upwards
- unified status console to highlight improper behavior, predict failure, and suggest corrective action
- Techniques: Stamp data blocks with modules used; Log faults, errors, failures and recovery methods

Lessons Learned from Other Fields

• 1800s: 25% railroad bridges failed!

• Techniques invented since:

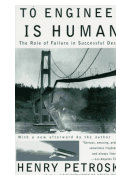
- Learn from failures vs. successes
- Redundancy to survive some failures
- Margin of safety 3X-6X times calculated load to cover what they don't know

• Safety now in Civil Engineering DNA

- "Structural engineering is the science and art of designing and making, with economy and elegance, structures that can safely resist the forces to which they may be subjected"

• Have we been building the computing equivalent of the 19th Century iron-truss bridges?

- What is computer equivalent of safety margin?



Recovery-Oriented Computing Conclusion

• New century needs new research agenda

- (and its not performance)

• Embrace failure of HW, SW, people and still build systems that work

• ROC: Significantly reducing Time to Recover/Repair

- => much greater availability
- + much lower maintenance costs



folklore, the Roc is known to be of such huge size that it can carry off elephants and other great land beasts with its large feet. Sinbad the Sailor encountered such a bird in The Thousand and One Nights.

Una Estrategia Alternativa para una Mala Carrera Académica

- **Advertencias:**
 - Desde el punto de vista de un líder de proyecto
 - Funciona para mí; no es la única forma
 - Principalmente desde una perspectiva académica en sistemas de cómputo
- **El objetivo es tener impacto:**
Cambiar la forma en que la gente hace Ciencia e Ingeniería de la Computación
 - Los académicos tienen un mal criterio de evaluación: artículos publicados
- **Seis pasos**
 - 1) Seleccionando un problema
 - 2) Escogiendo una solución
 - 3) Realizando un proyecto
 - 4) Terminando un proyecto
 - 5) Evaluación cuantitativa
 - 6) Transfiriendo tecnología

1) Seleccionando un Problema

¿Inventar un nuevo campo y permanecer en él?



- ¡No! Haz “cosas reales”: resuelve un problema que le interese a alguien
- ¡No! Usa proyectos separados y cortos
 - Siempre toman más tiempo
 - Se ajusta a la “vida” académica del estudiante
 - ¿Hacer un esfuerzo largo en un campo que cambia rápidamente?
 - Aprendiendo: Número de proyectos vs. tiempo de calendario
 - Si vas a fallar, mejor saberlo antes
- Esfuérzate por proyectos multi-disciplinarios con múltiples investigadores
 - 1 experto/área es ideal (no hay discusiones)
- Conjunta las fortalezas y debilidades del ambiente local
- Asegúrate de seguir interesado lo suficiente para trabajar en ello por 3-5 años
 - Los prototipos son útiles

Mi primer proyecto

- **Proyecto Multiprocesador con 3 académicos en hardware (“Xtree”)**
- **1977: Diseñar nuestro propio conjunto de instrucciones, microprocesador, topología de interconexión, ruteo, tarjetas, sistemas, sistema operativo**
- **Experiencia inmaculada:**
 - Ninguna en VLSI
 - Ninguna en microprocesadores
 - Ninguna en redes
 - Ninguna en sistemas operativos
- **Recursos inmaculados:**
 - Sin personal
 - Sin computadora dedicada (usamos la PDP-11/70 del departamento)
 - Sin herramientas CAD
 - Sin aplicaciones
 - Sin fondos
- **Resultados: 2 artículos de revista, 12 artículos de conferencia, 20 reportes técnicos**
- **¿Impacto?**

2) Escogiendo una solución

¿Dejarte guiar por la complejidad?

- **¡No!** Mantén todo simple a menos que haya una buena razón para no hacerlo
 - Escoge puntos de innovación con cuidado, y que sea compatible en todo lo demás
 - Los mejores resultados son obvios en retrospectiva
“Cualquiera pudo haber pensado eso”
- El costo de la complejidad hace mas largo el diseño, construcción, prueba, y verificación
 - Campo con cambios rápidos + retrasos
=> resultados menos impresionantes

¿Usar el Método Científico de la Computación?

- **¡No!** Realiza experimentos para descubrir problemas reales
- Usa tu intuición para **hacerte** preguntas, **no** para **responderlas**



(¡Y escoge un buen nombre!)

Reduced
Instruction
Set
Computers

Redundant
Array of
Inexpensive
Disks

Recovery
Oriented
Computing

...

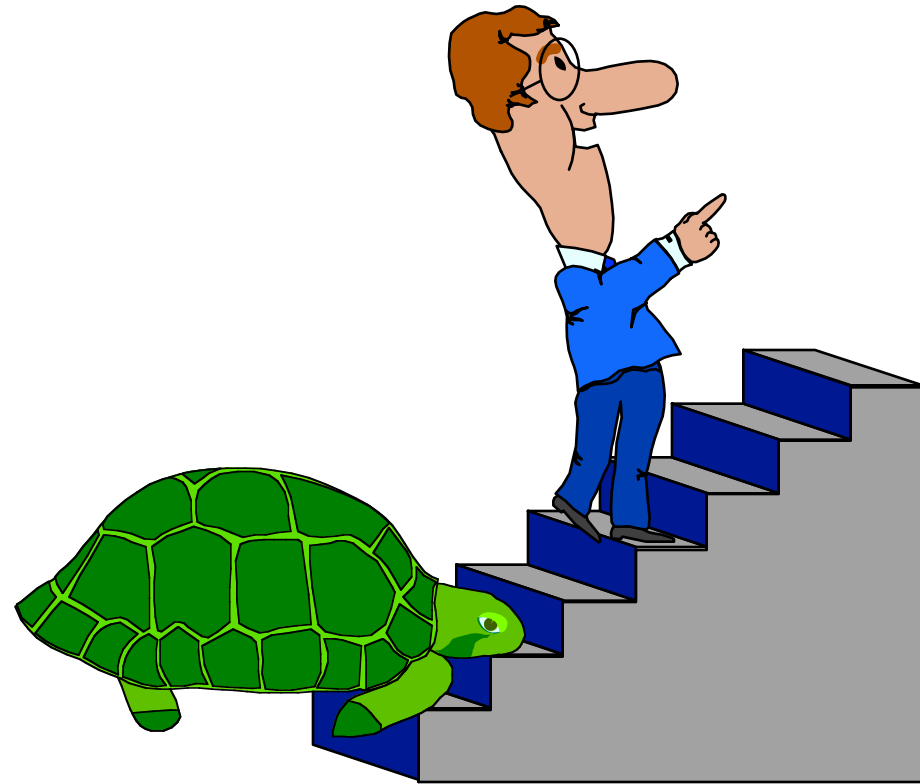
3) Realizando un proyecto



¿Evita la realimentación?

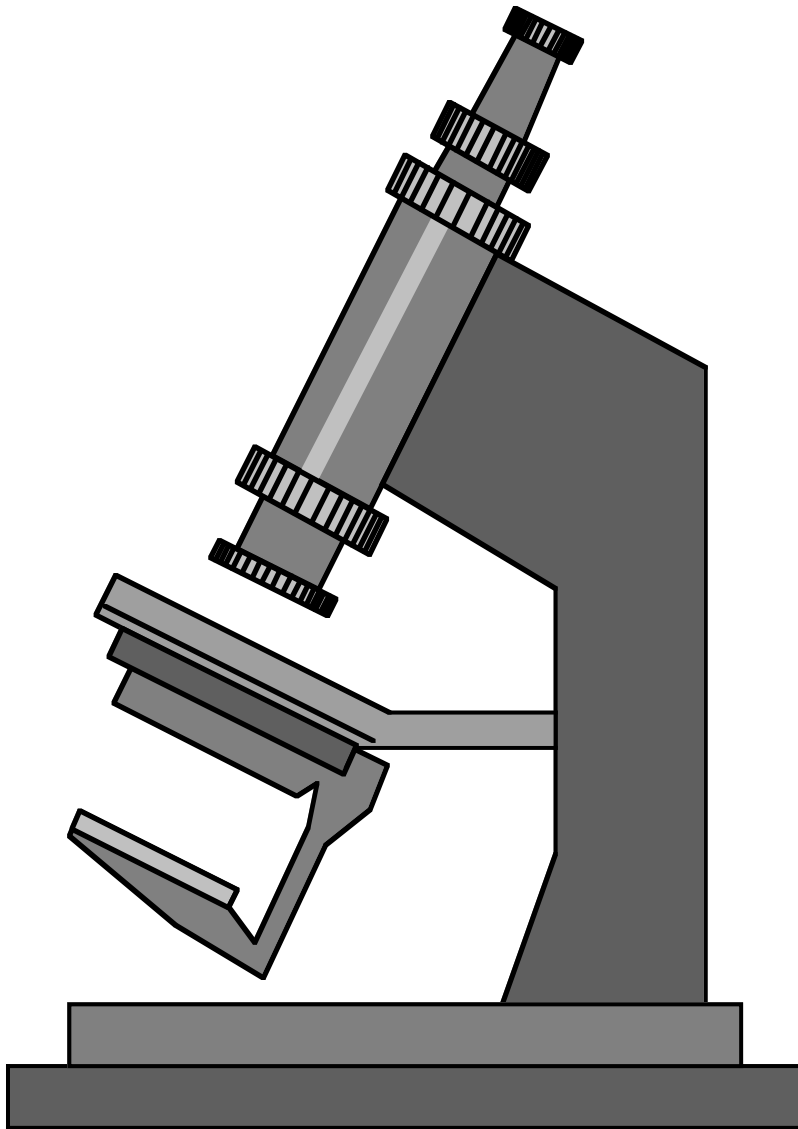
- ¡No! Haz revisiones periódicas del proyecto con personas de fuera
 - Dos veces al año: un retiro de 3 días
 - Académicos, estudiantes, personal + **invitados**
 - La pieza clave es la realimentación al final
 - Ayuda a generar plazos y espíritu de equipo
 - Da a los estudiantes oportunidad de dar muchas presentaciones, interactuar entre sí y con personas de la industria
- Considéralo como una corrección a medio curso
 - Un campo de cambio rápido + proyectos de 3-5 años => suposiciones han cambiado
- Escoge cuidadosamente el tamaño y a los miembros de tu equipo
 - Personalidades pesadas son difíciles para todos
 - De nuevo, 1 académico por área reduce la posibilidad de desacuerdo

4) Terminando un proyecto



- Las personas cuentan los proyectos que terminas, no los que comienzas
- Los proyectos exitosos llegan a una fase dura y poco atractiva
- Diseñar es más divertido que hacer el trabajo
 - “No hay ganadores en un equipo perdedor; no hay perdedores en un equipo ganador.”
 - “Puedes rápidamente decir si los autores han construido y echado a andar algo.”
- Reduce el proyecto si se retrasa
 - “Añadir gente a un proyecto retrasado lo retrasa más.”
- Terminar un proyecto es como la gente adquiere buen gusto para seleccionar buenos problemas, y encontrar soluciones simples

5) Evaluar cuantitativamente



¿Que nunca demuestren que estás equivocado?

- ¡No! Si no puedes demostrar que puedes estar equivocado, entonces tampoco puedes demostrar que estás en lo correcto
- Reporta en suficiente detalle para que otros puedan reproducir tus resultados
 - No puedes convencer a otros si no pueden obtener los mismos resultados
- Para bien o para mal, los criterios y resultados dan forma a un campo
- Los buenos aceleran el progreso
 - Buenos objetivos de desarrollo
- Los malos lastiman el progreso
 - ¿Ayudar a usuarios reales v. ayudar a las ventas?

6) Transfiriendo tecnología ¿Publicar artículos de revista ES Transferencia de Tecnología?



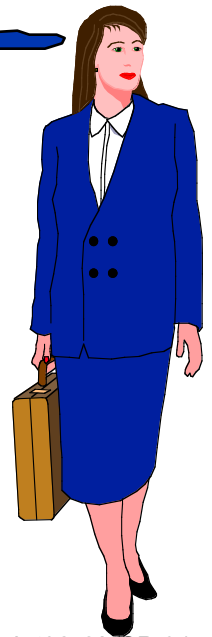
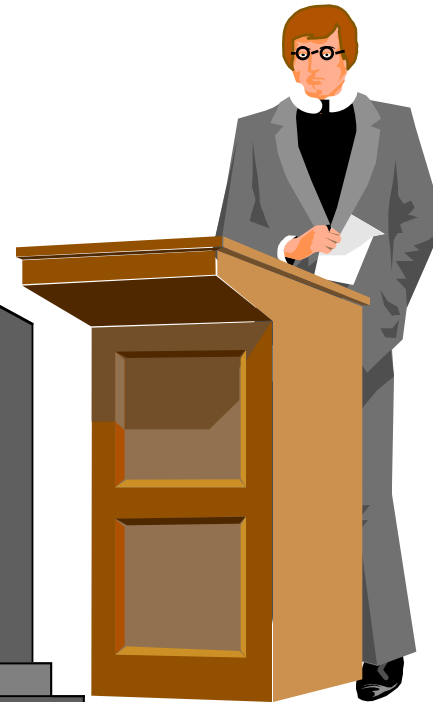
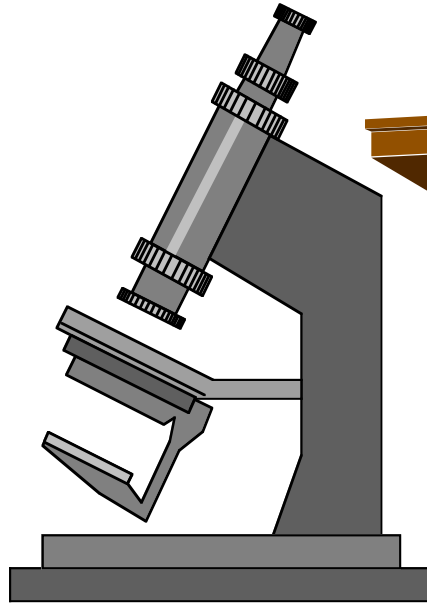
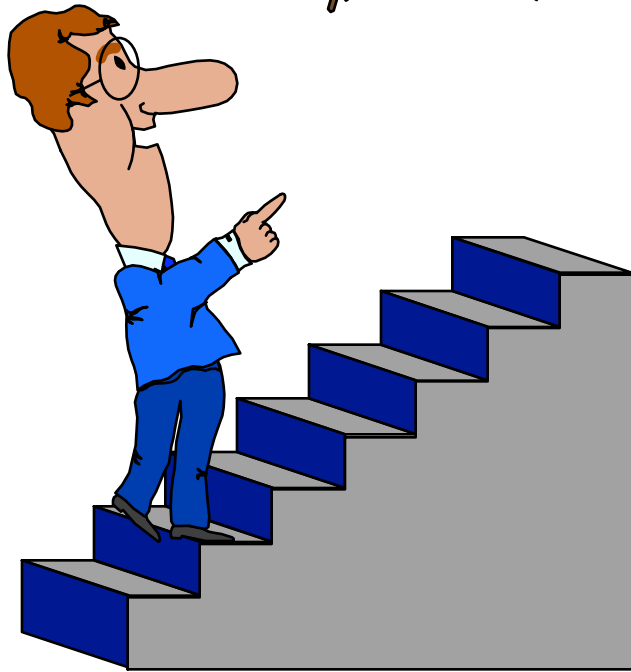
- ¡No! Haz trabajo de misionero: primero “Sermones”, después leerán tus artículos
 - Seleccionar un problema es la clave: haz “Cosas reales”
 - » Idealmente, hay más interés conforme pasa el tiempo
 - » Cambiarás la forma de pensar con resultados creíbles
 - » Las Prima Donnas interfieren con la transferencia
- Mi experiencia: la industria es reacia a realizar cambios
 - Howard Aiken, alrededor de 1950:
“¡El problema en este negocio no es hacer que la gente no robe tus ideas; es hacerlos que roben tus ideas!”
 - Se necesita 1 una compañía audaz (comúnmente no la No. 1) para tomar el riesgo **y** tener éxito
 - » RISC con Sun, RAID con (Compaq, EMC, ...)
 - Solo entonces, el resto de la industria te seguirá

6) Transfiriendo Tecnología



- **Pros**
 - Satisfacción personal: ver tu producto usado por otros
 - Ganancia Personal \$\$\$ (potencialmente)
 - Fama
- **Contras**
 - Aprender sobre planes de negocio, ventas vs. marketing, financiamiento, beneficios del personal, contrataciones, ...
 - Pasar más tiempo haciendo lo anterior vs. investigación y desarrollo
 - Solo 10% de las compañías realmente lo logran
 - Fama también si la compañía no tiene éxito (e.g., dot.com backlash)

Resumen: El paper del líder cambia durante el proyecto



Conclusión: Alternativas para una Mala Carrera Académica

- El objetivo es tener impacto:
Cambiar la forma en que la gente hace Ciencia e Ingeniería de la Computación
 - Muchos proyectos de 3 - 5 años dan más oportunidades de tener impacto
- La realimentación es la clave: busca y valora las críticas
- Haz “**cosas reales**”: asegúrate que estás resolviendo algún problema que le importa a alguien
- El buen criterio es crítico para seleccionar problemas de investigación, soluciones, experimentos, y comunicar resultados;
 - El buen criterio se adquiere con realimentación y proyectos concluidos
- El verdadero legado de la academia son las personas, no los artículos:
 - crear ambientes que desarrollen profesionales de los que estemos orgullosos
- *Los estudiantes* son la moneda de cambio en el reino académico