

# Pragmatismo en la Planificación de Proyectos

Diseño de software para cómputo científico

2020

# Anecdotas interesantes

## Mariner 1

- ▶ Fue la primera misión NASA del Programa Mariner en intentar sobrevolar Venus.
- ▶ A los 4 min y 53 s del lanzamiento se produjo una inclinación inesperada del cohete hacia el noreste que daba alguna posibilidad de que el cohete cayese cerca de las rutas transatlánticas,
- ▶ Se decidió autodestruir el cohete.
- ▶ **Problemas:** Se omitió un guion (–) en las instrucciones del programa de guiado del cohete.

# Anecdotas interesantes

## Mariner 1

- ▶ Fue la primera misión NASA del Programa Mariner en intentar sobrevolar Venus.
- ▶ A los 4 min y 53 s del lanzamiento se produjo una inclinación inesperada del cohete hacia el noreste que daba alguna posibilidad de que el cohete cayese cerca de las rutas transatlánticas,
- ▶ Se decidió autodestruir el cohete.
- ▶ **Problemas:** Se omitió un guion (–) en las instrucciones del programa de guiado del cohete.

## Resultado

Perdida de US\$18.5 million en 1962.

# Anecdotas interesantes

## Pentium FDIV bug

- ▶ Thomas Nicely, profesor de matemáticas en Lynchburg College, escribió un código, matemático el cual comenzó a mostrar inconsistencia el 13 de junio de 1994.
- ▶ Se logro demostrar que esto se debía a un error en el manejo de división de flotantes en los nuevos procesadores Pentium.
- ▶ El error de la división estaba en el orden de 61 partes por millón.
- ▶ El problema era una tabla de división a nivel ensamblador que le faltaba 5 registros sobre 1000.
- ▶ Intel reconoce posteriormente el bug, y si una persona podía demostrar que había sido afectada por el error, entonces Intel procedería a cambiar su procesador.
- ▶ El error era realmente poco importante para la mayoría.
- ▶ Los demandaban el reemplazo de los procesadores defectuosos y empresas como IBM se unieron a la denuncia.
- ▶ Intel se vio forzada a aceptar sustituir todos los microprocesadores defectuosos.

## Anecdotas interesantes

### Pentium FDIV bug

- ▶ Thomas Nicely, profesor de matemáticas en Lynchburg College, escribió un código, matemático el cual comenzó a mostrar inconsistencia el 13 de junio de 1994.
- ▶ Se logro demostrar que esto se debía a un error en el manejo de división de flotantes en los nuevos procesadores Pentium.
- ▶ El error de la división estaba en el orden de 61 partes por millón.
- ▶ El problema era una tabla de división a nivel ensamblador que le faltaba 5 registros sobre 1000.
- ▶ Intel reconoce posteriormente el bug, y si una persona podía demostrar que había sido afectada por el error, entonces Intel procedería a cambiar su procesador.
- ▶ El error era realmente poco importante para la mayoría.
- ▶ Los demandaban el reemplazo de los procesadores defectuosos y empresas como IBM se unieron a la denuncia.
- ▶ Intel se vio forzada a aceptar sustituir todos los microprocesadores defectuosos.

### Resultado

Perdida equivalente a \$730 millones de dólares en 2018.

# Anecdotas interesantes

## Mars Climate Orbiter

- ▶ Fue una sonda de la NASA lanzada desde Cabo Cañaveral el 11 de diciembre de 1998.
- ▶ Después de un viaje de 9 meses y medio, se estrelló contra Marte.
- ▶ La misión en tierra enviaba los datos en formato imperial, los instrumentos de NASA trabajan con sistema métrico.

## Resultado

El costo de la misión perdida fue de \$ 327,6 millones en total. El dato que encontré que la perdida fue de 125 millones de dólares (El cohete llevaba 2 sondas).

# Anécdotas interesantes

## Boeing 737 MAX

- ▶ En 2010, Airbus lanzó el A320neo, que incorporaba una nueva planta motriz más eficiente y presentaba unos costes operativos menores (7% mas eficiencia en combustible).
- ▶ Boeing aprueba en 2011 Aunque se calculó que diseñar un nuevo 737 que iba a tomar 6 años, pero estuvo listo en menos de tres meses.
- ▶ Boeing afirma que el 737 MAX consume un 16 % menos que los A320, y un 4 % menos que el A320neo.
- ▶ Este poco tiempo llevo a un error de software en los sistemas de navegación y simuladores.

# Anécdotas interesantes

## Boeing 737 MAX

- ▶ En 2010, Airbus lanzó el A320neo, que incorporaba una nueva planta motriz más eficiente y presentaba unos costes operativos menores (7% mas eficiencia en combustible).
- ▶ Boeing aprueba en 2011 Aunque se calculó que diseñar un nuevo 737 que iba a tomar 6 años, pero estuvo listo en menos de tres meses.
- ▶ Boeing afirma que el 737 MAX consume un 16 % menos que los A320, y un 4 % menos que el A320neo.
- ▶ Este poco tiempo llevo a un error de software en los sistemas de navegación y simuladores.

## Resultados

- ▶ Entre octubre de 2018 a marzo de 2019 dos 737 MAX 8 recién estrenados sufrieron accidentes fatales, los vuelos 610 de Lion Air y 302 de Ethiopian Airlines.
- ▶ 6 mil millones en pérdida.
- ▶ 346 muertos.



## Anécdotas interesantes

Pero claro: A mí no me va a pasar.

## Anécdotas interesantes

Pero claro: A mí no me va a pasar.

- ▶ Desde 2016 mantengo el paquete Scikit-Criteria.
- ▶ Es una colección de Métodos de Análisis Multi-Criterio Integrado Al Stack Científico de Python.
- ▶ Lo hice por hobby, y esta en un estado de beta constante.
- ▶ El año pasado me llega este mail.

## Anécdotas interesantes

[Redacted]

23 sept. 2019 2:16



para mí ▾



inglés ▾



español ▾

[Traducir mensaje](#)

[Desactivar para: inglés](#) ×

Greetings to the development and support team,

This is an appreciation mail. My name is [Redacted] and I'm involved in developing cost-effective and prioritization models for my firm . I recently came across your scikit criteria library and it was extremely helpful in designing my model.

You guys have done a great job and I'm really thankful for them. All the best for your future endeavors.

## Anécdotas interesantes

Pero claro: A mí no me va a pasar.

- ▶ Desde 2016 mantengo el paquete Scikit-Criteria.
- ▶ Es una colección de Métodos de Análisis Multi-Criterio Integrado Al Stack Científico de Python.
- ▶ Lo hice por hobby, y esta en un estado de beta constante.
- ▶ El año pasado me llega este mail.
- ▶ Y ya que estamos vamos a scholar

<https://scholar.google.com/citations?user=qhWYqpoAAAAJ&hl=en>

## Consideraciones:

1. Toda respuesta profesional empieza con la palabra “**depende**”
2. Tener en cuenta que error, falla y no-conformidades no son lo mismo.
3. Overkill



# Agenda

- ▶ Ingeniería de Software.
- ▶ Proyectos (Triangulo de Hierro).
- ▶ Ciclo de vida de un proyecto
- ▶ Que es un objetivo (SMART).
- ▶ Estimación de tiempos (PERT).
- ▶ Complejidad de tareas.



## Ingeniería de Software

*La ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación, y mantenimiento del software.*

## Software

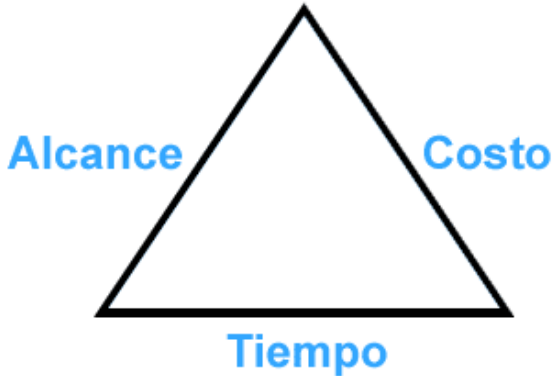
*Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones de un sistema de computación.*

IEEE Standard Glossary  
of Software Engineering Terminology

# Proyectos

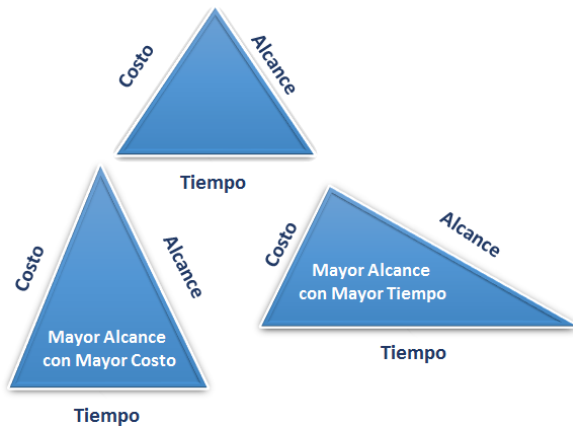
It's a temporary endeavor undertaken to create a unique product, service or result. (PMI)

## Triángulo de Hierro



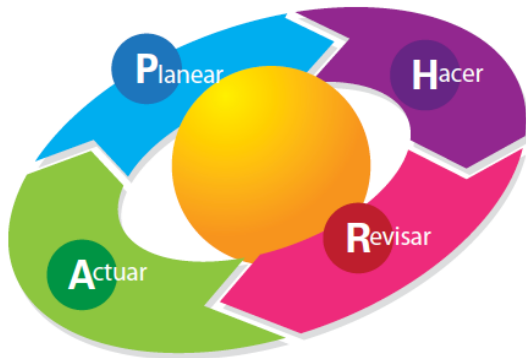


# Proyectos



# Ciclo de Deming

---



- Solo nos fijamos un poco en Planear y Revisar

# Objetivos

## Según la teoría general de sistemas

*El elemento programático que identifica la finalidad hacia la cual deben dirigirse los recursos y esfuerzos para dar cumplimiento a los propósitos.*



© Scott Adams, Inc./Dist. by UFS, Inc.

# Objetivos - SMART

- ▶ **Specific** – target a specific area for improvement.
- ▶ **Measurable** – quantify or suggest an indicator of progress.
- ▶ **Achievable** – can be realistically achieved, given.
- ▶ **Responsible** – specify who will do it.
- ▶ **Time-related** – specify when the result(s) can be achieved.



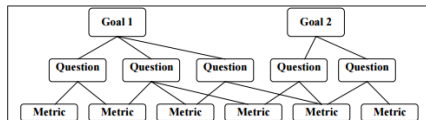
## Objetivo - SMART

- ▶ Voy adelgazar. (No Smart)
- ▶ Voy a adelgazar 100 Kg, en un mes usando reduce-fat-fast. (**SMART**)



# Métricas

- ▶ No se puede controlar lo que no se puede medir.
- ▶ El enfoque que se usa es GQM el cual deriva Objetivos (**G**) a preguntas (**Q**) las cuales tratan de responderse con métricas (**M**).



Goal	Purpose Issue Object (process) Viewpoint	Improve the timeliness of change request processing from the project manager's viewpoint
Question		What is the current change request processing speed?
Metrics		Average cycle time Standard deviation % cases outside of the upper limit
Question		Is the performance of the process improving?
Metrics		$\frac{\text{Current average cycle time}}{\text{Baseline average cycle time}} * 100$ Subjective rating of manager's satisfaction

## Estimación de Tiempos y tareas criticas (PERT)

- ▶ Preguntas razonables cuando uno plantea una idea:
  - ▶ Como lo vas a hacer?
  - ▶ Quien lo va a hacer?
  - ▶ Cuanto vas a tardar? (**y aca morimos todos**)
- ▶ PERT (del inglés, **P**roject **E**valuation and **R**evuew **T**echniques), son un modelo para la administración y gestión de proyectos inventado en 1957 por la Oficina de Proyectos Especiales de la Marina de Guerra del Departamento de Defensa de EE.UU. como parte del proyecto Polaris de misil balístico móvil lanzado desde submarino. Este proyecto fue una respuesta directa a la crisis del Sputnik.

## PERT (cont.)

En planificación y programación de proyectos se estima que la duración esperada de una actividad es una variable aleatoria de distribución de probabilidad Beta Unimodal:

$$t_e = \frac{(t_o + 4t_m + t_p)}{6}$$

- ▶  $t_e$ : Expected time
- ▶  $t_o$ : the minimum possible time required to accomplish a task, assuming everything proceeds better than is normally expected
- ▶  $t_p$ : the maximum possible time required to accomplish a task, assuming everything goes wrong but excluding major catastrophes.
- ▶  $t_m$ : the best estimate of the time required to accomplish a task, assuming everything proceeds as normal.



## PERT (cont.)

La desviación estandar de la tarea esta dada por:

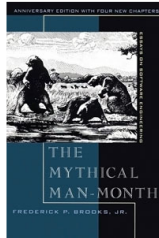
$$\sigma = \frac{t_p - t_o}{6}$$

### Ejemplos

- ▶ [https://github.com/leliel12/otree\\_korbinian/blob/master/\\_doc/estimation.ipynb](https://github.com/leliel12/otree_korbinian/blob/master/_doc/estimation.ipynb)
- ▶ [https://github.com/leliel12/otree\\_wissink/blob/master/\\_doc/estimation.ipynb](https://github.com/leliel12/otree_wissink/blob/master/_doc/estimation.ipynb)

# Frederick Brooks

Turing Award 199: For landmark contributions to computer architecture, operating systems, and software engineering.



## Frederick Brooks (cont)

### The Mythical Man-Month

Complex programming projects cannot be perfectly partitioned into discrete tasks that can be worked on without communication between the workers and without establishing a set of complex interrelationships between tasks and the workers performing them. Therefore, assigning more programmers to a project running behind schedule will make it even later.

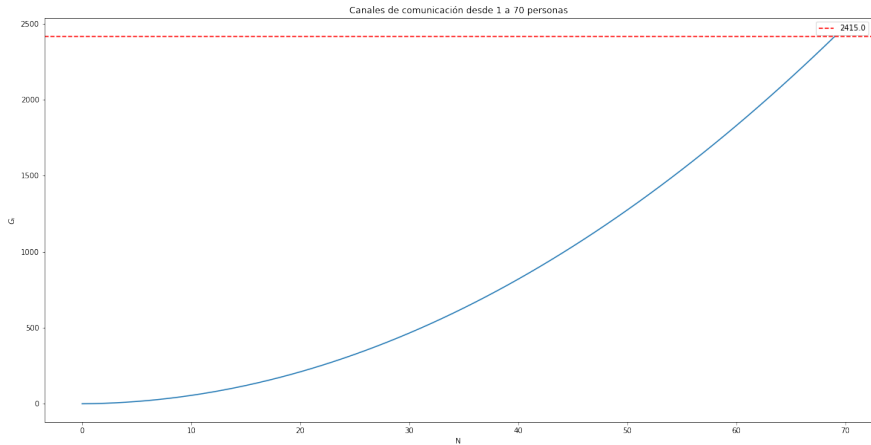
### Group intercommunication formula

$$G_i = \frac{n(n-1)}{2}$$

**Example:** 50 personas

$$1225 = \frac{50(50-1)}{2}$$

## Group intercommunication formula



## Frederick Brooks (cont)

### No-Silver Bullets

There is no single development, in either technology or management technique, which by itself promises even one order of magnitude improvement within a decade in productivity, in reliability, in simplicity.



## Frederick Brooks (cont)

### The tendency towards irreducible number of errors

In a suitably complex system there is a certain irreducible number of errors. Any attempt to fix observed errors tends to result in the introduction of other errors.

99 little bugs in the code.  
99 little bugs.  
Take one down, patch it around.  
127 little bugs in the code...



# Frederick Brooks (cont)

## Progress tracking

- ▶ **Question:** How does a large software project get to be one year late?
- ▶ **Answer:** One day at a time!"



# Frederick Brooks (cont)

## Conceptual integrity

To ensure a user-friendly system, a system may deliberately provide fewer features than it is capable of. The point is that, if a system is too complicated to use, then many of its features will go unused because no one has the time to learn how to use them.

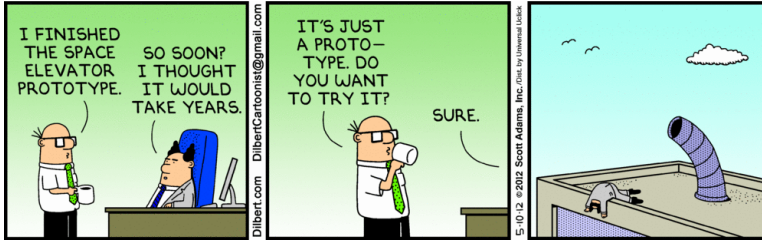




# Frederick Brooks (cont.)

## The pilot system

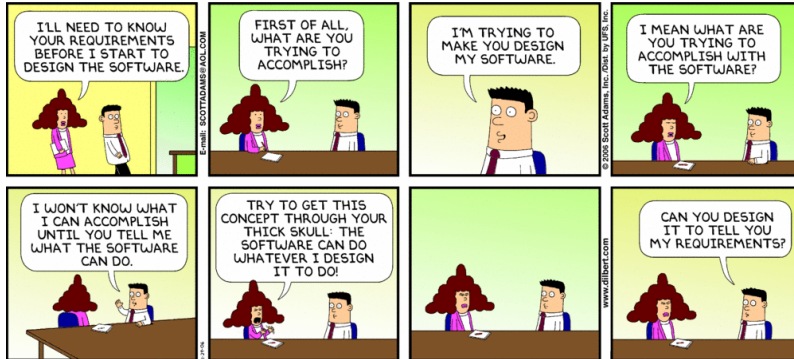
When designing a new kind of system, a team will design a throw-away system (whether it intends to or not).



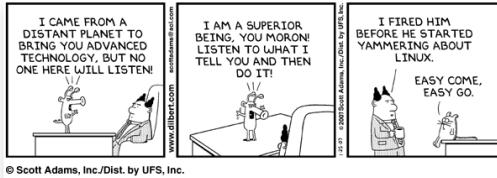
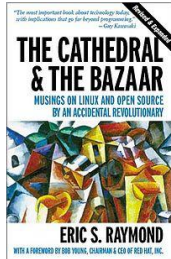
# Frederick Brooks (cont.)

## Code freeze

Software is invisible. Therefore, many things only become apparent once a certain amount of work has been done on a new system, allowing a user to experience it. This experience will yield insights, which will change a user's needs or the perception of the user's needs.



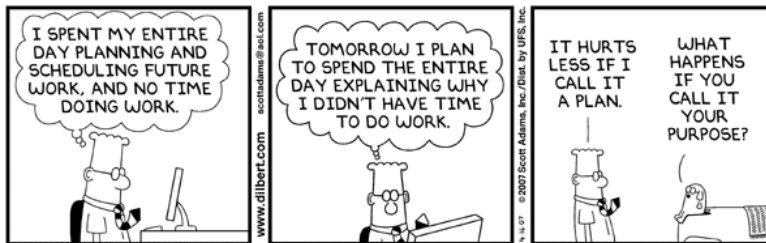
# Eric Raymond



## Eric Raymond (cont.)

- ▶ Given enough eyeballs, all bugs are shallow (linus' law)
- ▶ Good programmers know what to write. Great ones know what to rewrite (and reuse)
- ▶ Release early. Release often. And listen to your customers.
- ▶ Often, the most striking and innovative solutions come from realizing that your concept of the problem was wrong.
- ▶ Perfection (in design) is achieved not when there is nothing more to add, but rather when there is nothing more to take away

# Conclusiones & Preguntas



## Slides

[https://github.com/leliel12/talks/blob/master/iate2017/proyectos\\_sem/slides.pdf](https://github.com/leliel12/talks/blob/master/iate2017/proyectos_sem/slides.pdf)

## Fuentes

- ▶ [https://en.wikipedia.org/wiki/Mariner\\_1](https://en.wikipedia.org/wiki/Mariner_1)
- ▶ [https://en.wikipedia.org/wiki/Pentium\\_FDIV\\_bug](https://en.wikipedia.org/wiki/Pentium_FDIV_bug)
- ▶ <https://www.youtube.com/watch?v=6FsmLwUITV4&list=PLFQMOORsd0x5pRstKFQJvmdkqCpWDR7YP&index=9&t=0s>
- ▶ [https://es.wikipedia.org/wiki/Boeing\\_737\\_MAX](https://es.wikipedia.org/wiki/Boeing_737_MAX)