

אודות הקשר האגדי  
בין אנשים וחודשים  
(רמז: לא מה שחשבתם)

**The Mythical Man-  
Month**

Essays on Software  
Engineering,

Frederick P. Brooks, Jr.

University of North  
Carolina at Chapel Hill

פיתוח מערכות תוכנה מבוססות Java  
בית הספר למדעי המחשב, אוניברסיטת תל אביב



# Frederick P. Brooks, Jr.

---

נולד ב 1931

התחיל את דרכו כאיש תוכנה

IBM System/360 1956-1963 ארכיטקט חומרה

IBM System/360 של האבא

Alphabetical Read-Out Device

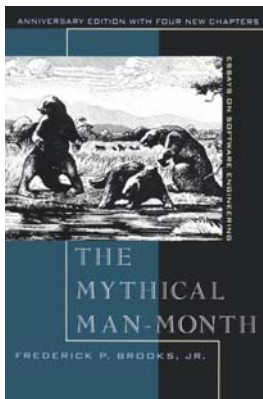
Program Interruption System

ארכיטקטורה

1964-1965 מונה למנהל Operating System/360 (MVS/370 כיום)

1964-1984 ייסד את החוג למדעי המחשב באוניברסיטת צפון קרוליינה (Chapel Hill) ועמד בראשו

כיום עוסק בהוראת תוכנה ובמחקר virtual environments



# The Mythical Man-Month

□ אוסף מאמרים

□ נכתב כאוסף תובנות שנצברו בתקופת היותו מנהל הפיתוח של מערכת ההפעלה OS/360

□ וכן שיחות עם מנהלי פרויקטי תוכנה גדולים בתעשייה ובאקדמיה

□ פורסם לראשונה ב 1975

□ מהדורת יובל ה-20 שפורסמה ב 1995 כוללת את המאמרים המקוריים, את המאמר "No Silver Bullet" (שהתפרסם לראשונה ב 1986), וכן דיון לגבי תקפות הטענות שהופיעו במאמרים המקוריים



# The Tar Pit

---

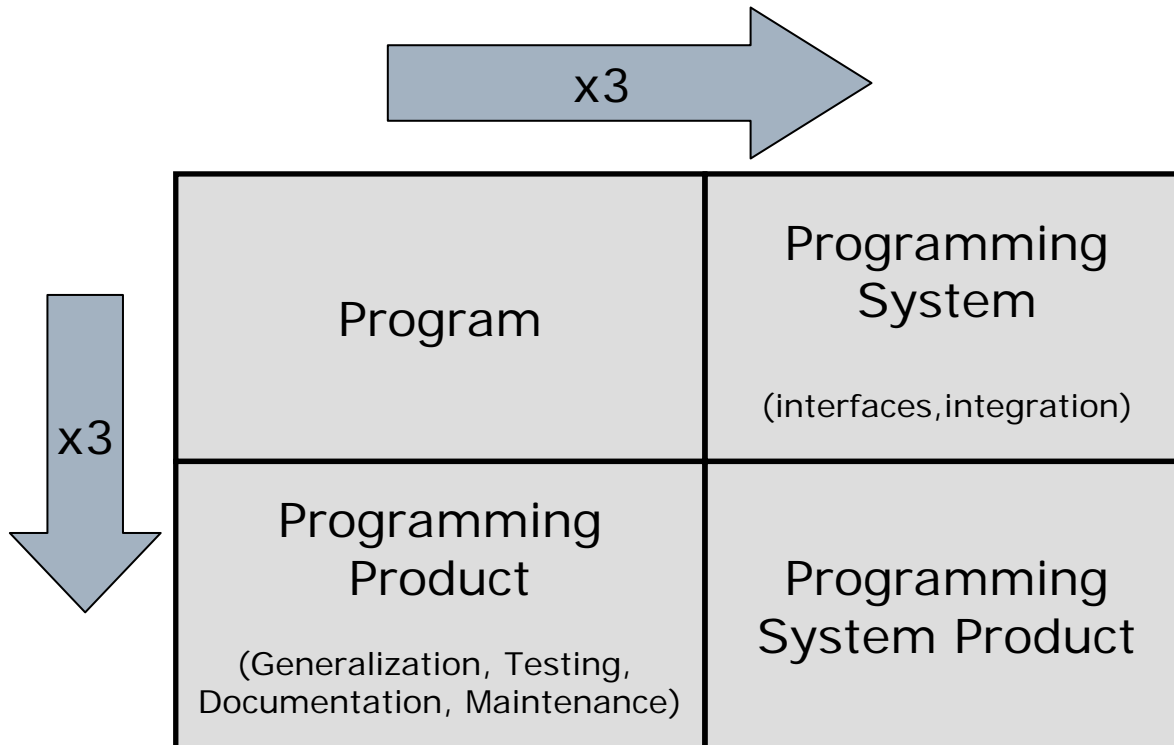
- *Een schip op het strand is een baken in zee.*  
*[A ship on the beach is a lighthouse to the sea. ]*

***DUTCH PROVERB***

- האגדה האורבנית על זוג מתכנתים וגראז'
  - התפוקה בתעשייה היא של 1000 statements/year למתכנתת בצוות
  - מה מקור הפער?

# החלום ושברו

---





# אושר מהול בצער

- 
- |  |  |
|--|--|
| <input type="checkbox"/> שלמות                 | <input type="checkbox"/> חדוות היצירה      |
| <input type="checkbox"/> חוסר שליטה/עצמאות     | <input type="checkbox"/> עשיית דבר שימושי  |
| <input type="checkbox"/> תלות בביצועי עמיתים   | <input type="checkbox"/> מורכבות היא מרתקת |
| <input type="checkbox"/> נוקדנות               | <input type="checkbox"/> למידה             |
| <input type="checkbox"/> תפוקה לינארית או פחות | <input type="checkbox"/> יצירתיות מופשטת   |
| <input type="checkbox"/> הדבר הבא              |  |

# Restaurant Antoine

Fondé En 1840

## AVIS AU PUBLIC

Faire de la bonne cuisine demande un certain temps. Si on vous fait attendre, c'est pour mieux vous servir, et vous plaire.

ENTREES (SOUFFLE)	
Côtelettes d'agneau grillées 2.50	Entrecôte marchand de vin 4.00
Côtelettes d'agneau aux champignons frais 2.75	Côtelettes d'agneau maison d'or 2.
Filet de boeuf aux champignons frais 4.75	Côtelettes d'agneau à la parisienne
Ris de veau à la financière 2.00	Fois de volaille à la brochette 1.50
Filet de boeuf nature 3.75	Tournedos nature 2.75
Tournedos Médicis 3.25	Filet de boeuf à la hawaïenne 4.00
Pigeonneaux sauce paradis 3.50	Tournedos à la hawaïenne 3.25
Tournedos sauce béarnaise 3.25	Tournedos marchand de vin 3.25
Entrecôte minute 2.75	Pigeonneaux grillés 3.00
Filet de boeuf béarnaise 4.00	Entrecôte nature 3.75
Tripes à la mode de Carn (commander d'avance) 2.00	Châteaubriand (30 minutes)

LÉGUMES	
Epinards sauce crème .60	Chou fleur au gratin .60
Broccoli sauce hollandaise .80	Asperges fraîches au beurre .90
Pommes de terre au gratin .60	Carottes à la crème .60
Haricots verts au beurre .60	Pommes de terre soufflées
Petits pois à la française .75	

SALADES	
Salade Antoine .60	Fonds d'artichauts Bayard
Salade Mirabeau .75	Salade de laitue aux peufs .60
Salade laitue au roquefort .80	Tomate frottée à la Jules César .60
Salade de laitue aux tomates .60	Salade de coeur de palmier 1.00
Salade de légumes .60	Salade aux pointes d'asperges .60
Salade d'anchois 1.00	Avocat à la vinaigrette .60

DESSERTS	
Gâteau moka .50	Cerises jubilé 1.25
Meringue glacée .60	Crêpes à la gelée .60
Crêpes Suzette 1.25	Crêpes nature .70
Glace sauce chocolat .60	Omelette au thon 1.10
Fruits de saison à l'eau-de-vie .75	Glacé à la vanille .60
Omelette soufflée à la Jules César (2) 2.00	Frisoles au kirsch
Omelette Alaska Antoine (2) 2.50	Pêche Meiba

FROMAGES	
Roquefort .50	Liederkranz .50
Camembert .50	Gruyère .50
	Fromage à la crème Philadelphie .50

CAFÉ ET THÉ	
Café .20	Café au lait .20
Café brûlé diabologique 1.00	Thé glacé .20
	Thé .20
	Demi-tasse

EAUX MINÉRALES—BIÈRE—CIGARES—CIGARETTES	
White Rock	Bière locale
Vichy	Canada Dry
Cliquot Club	Cigarettes

Roy E. Alciatore, Propriétaire

713-717 Rue St. Louis Nouvelle Orléans, Louisiane

## The Mythical Man-Month

- *Good cooking takes time. If you are made to wait, it is to serve you better, and to please you.*

**MENU OF RESTAURANT ANTOINE, NEW ORLEANS**

□ סיבות להערכות זמנים לקויות:

■ אופטימיות

■ מאמץ אינו התקדמות

■ בטחון עצמי

■ ניטור לקוי

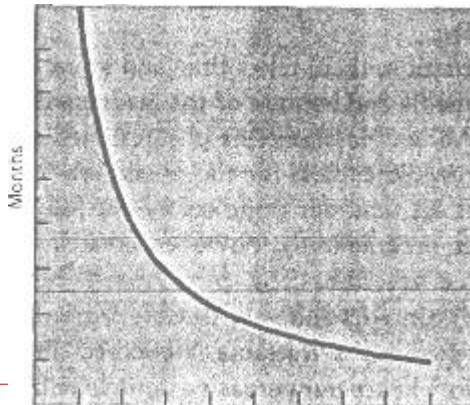
■ הוספת כ"א לפרויקט מאחר

# The Mythical Man-Month

"... the man-month as a unit for measuring the size of a job is a dangerous and deceptive myth. It implies that men and months are interchangeable"

ניתן להמיר כח אדם בזמן רק בפעולות שאינן דורשות תקשורת בין האנשים (איסוף כותנה, קציר חיטה)

תכנות מערכות אינו כזה



Men



# חלוקת המשימה

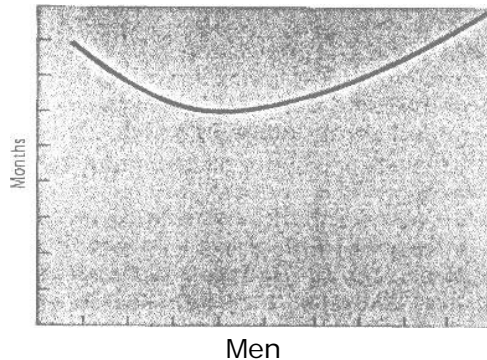
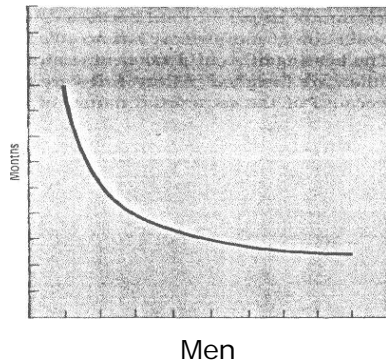
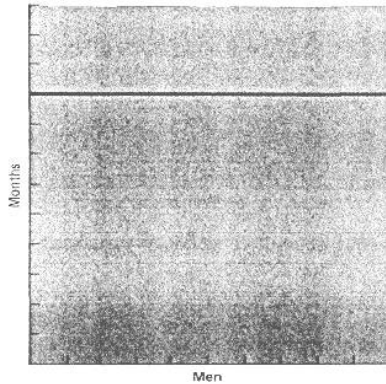
טבעו של ניפוי השגיאות

גם כאשר משימה ניתנת לחלוקה – לתקשורת יש תקורה:

■ הכשרה

■ תאום  $(n(n-1)/2)$

וכאשר התאום מורכב:



# הערכות זמנים לפרויקט

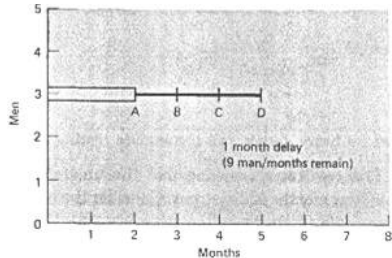
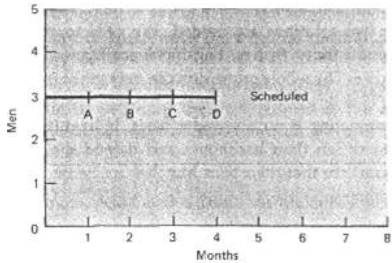
כלל אצבע:

- 1/3 planning
- 1/6 coding
- 1/4 component test and early system test
- 1/4 system test, all components in hand

דוגמא:

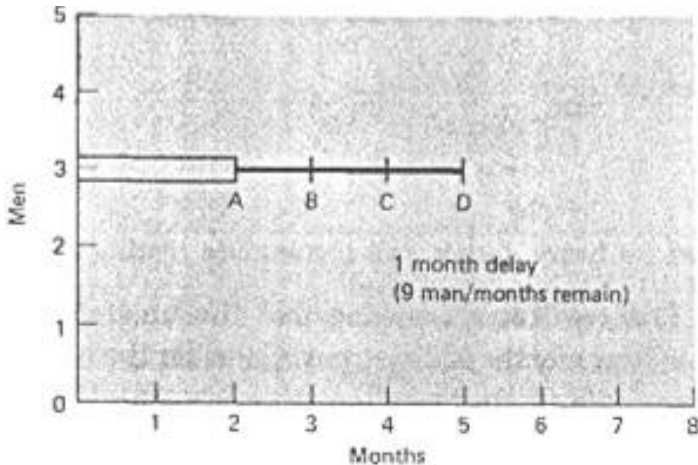
- משימה הוערכה ב 12 חודשי אדם
- למשימה הוקצו 3 מתכנתות ל-4 חודשים
- הוגדרו 4 אבני דרך (A, B, C, D) בסופו של כל חודש

- נניח כי רק לאחר חודשיים הושלמה אבן הדרך הראשונה
- מה ניתן לעשות? נציג כמה חלפות



# אפשרות א'

- ההנחה היא שהמשימה חייבת להסתיים בזמן
- רק החלק הראשון הובן שלא כהלכה
- נשארו למשימה 9 חודשי אדם ורק חודשיים קלנדריים
- לשם כך דרושים 4.5 אנשים



- מסקנה:
- נוסף 2 מתכנתות ל-3 הקיימות

# אפשרות ב'

□ ההנחה היא שהמשימה חייבת להסתיים בזמן

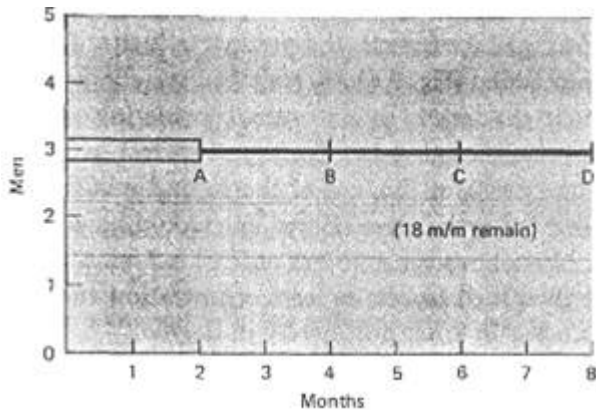
□ הערכות הזמנים היו נמוכות בצורה אחידה

□ בעצם נשאר למשימה 18 חודשי אדם ורק חודשיים קלנדריים

□ לשם כך דרושים 9 אנשים

□ מסקנה:

■ נוסף 6 מתכנתות ל-3 הקיימות



# אפשרויות נוספות

---

אפשרות ג'

- Reschedule
- "Take no small slips"

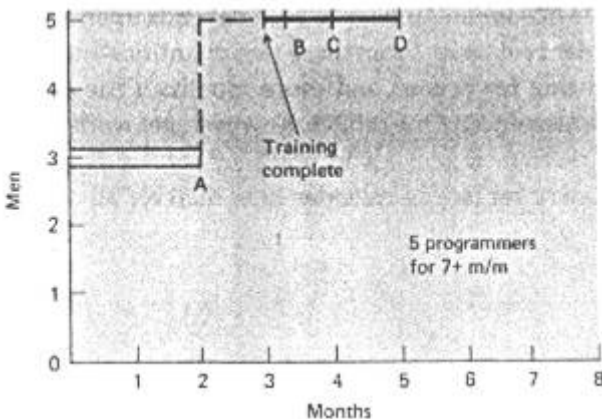
אפשרות ד'

■ קיצוץ המשימה

על פניו נראה כאילו שתי האפשרויות האלה פחות טובות משתי האפשרויות הקודמות – האמנם?

# ההשלכות של אפשרות א'

- גם אם נזניח את גיוס 2 המתכנתות המוכשרות הנוספות
- עדיין תידרש אחת המתכנתות הוותיקות להכשיר אותן
- אפילו הכשרה של חודש קלנדרי אחד עולה 3 חודשי אדם (מסך חודשי האדם הדרושים בתכנון החדש)
- את המשימה שפוצלה במקור ל-3, יש לפצל עכשיו ל-5
- מאמץ שכבר הושקע בעבודה שהושלמה עשוי להתבזבז במהלך
- שלב בדיקות האינטגרציה ובדיקות המערכת מסתבך
- בחלוף החודש השלישי עדיין נותרו 7 חודשי-אדם של עבודה ורק חודש קלנדרי אחד
- העיכוב המקורי נשאר – עלות הפרויקט עלתה



# the plot thicken...

---

- אז אולי פשוט היינו צריכים להוסיף 4 מתכנות? (ולא רק 2)
- נשים לב כי בכל מקרה של הוספת כח אדם עדיין אין עמידה באבן הדרך הבאה – מה שעלול להוביל לחזרה רקורסיבית על האלגוריתם...
- כל זאת תחת ההנחה קרתה רק באבן הדרך הראשונה...

Brooks's Law:

*Adding manpower to a late software project makes it later*

# No Silver Bullet – Essence and Accident in Software Engineering (1)

---

- *“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” (1986).*

**השקפים הבאים לקוחים ממצגת של יורי ארביטמן ממכון ויצמן למדע** □



# No Silver Bullet – Essence and Accident in Software Engineering

---

- Silver bullet: a way to defeat werewolves.
  - Generally [in folklore]: any straightforward solution perceived to have extreme effectiveness.
  
- Compares software to hardware:
  - The anomaly is not that software progress is so slow, but that computer hardware progress is so fast.
  
- **Essence**—the difficulties inherent in the nature of the software
- **Accidents**—those difficulties that today attend its production but that are not inherent [but *incidental*].

# Essence Difficulties

---

- **Essence:**
  - Complexity
  - Conformity
  - Changeability
  - Invisibility

# Essence Difficulties

---

## *Complexity*

- enormous number of states (orders of magnitude more than in hardware), so conceiving, describing and testing is hard
- increases non-linearly with its size
- introduces a lot of difficulties:
  - communication among team members
  - enumerating (much less understanding) of all possible states of the program
  - management problems:
    - conceptual integrity is hard to achieve
    - learning curve: personnel turnover becomes disaster
  - others

# Essence Difficulties

---

## □ *Conformity*

- Physics example: looking for simplicity in complex structures
- Software: the complexity is arbitrary, forced by existing systems to which the interfaces must conform.
  - cannot be simplified by any redesign!

# Essence Difficulties

---

## *Changeability*

- Software is constantly under pressure for change, partly because it can be changed more easily than a building.
- Two processes are at work:
  - Demand for extended function (a result of success)
  - Suitability for a new hardware is needed

# Essence Difficulties

---

## □ *Invisibility*

- Unlike other disciplines, where geometric abstractions serve as a powerful tool, software is not inherently embedded in space
- Several general directed graphs, superimposed one upon another appear while trying to create a representation
  - the graphs are nor planar neither hierarchical

# Hopes for the silver

---

- What helped to overcome some of accidental difficulties in the past?
  - High-level languages
  - Unified programming environments
  
- Hopes for the silver:
  - OOP:
    - Hierarchical
    - Data hiding

*Helps in design, but do not solve design complexity problem*

# Hopes for the silver

---

- AI (expert systems)
  - *May be very useful*
- AI ("Automatic programming"): generation of a program from problem specification
  - Used successfully for very specific tasks (differential equations,...)
  - Hard to imagine having a general solution
- Graphical programming:
  - No hope, for software is difficult to visualize



# Hopes for the silver

---

- Program verification
  - Might reduce the program-testing load, not eliminate it
  - A lot of work
  - Can establish that a program meet its specification. But the hardest part is to get such complete and consistent specification!
  
- Better workstations, environments and tools
  - are welcomed, but magical enhancements cannot be expected

# Addressing Essence

---

- Buy vs. Build
  - Discusses the process of wide-spread use of software “today” compared to 60-s, adopting procedures to existing software
  
- Requirements refinement and rapid prototyping
  - *“The hardest single part of building a software system is deciding precisely what to build”*
  - Thus, rapid prototyping tools are one of the most promising efforts that attack *the essence* of software development problem.

# Addressing Essence

---

- Incremental development
  - Write vs. Build
  - Build vs. Grow (top-down design, stubs...)
- Great designers
  - *“The difference between the great and the average approach an order of magnitude”*
  - Gives hints as to how to grow great designers