

# ICS 121 Lecture Notes

## The Mythical Man-Month Frederick P. Brooks, Jr.

Topic 11  
Mythical Man-Month 1

ICS 121

- Published 1975, Republished 1995
- Experience managing the development of OS/360 in 1964-65
- Central Argument
  - Large programming projects suffer management problems different in kind than small ones, due to division of labor.
  - Critical need is the preservation of the conceptual integrity of the product itself.
- Central Conclusion
  - Integrity achieved through exceptional designer.
  - Implementation achieved through well-managed effort.

## The Tar Pit

Topic 11  
Mythical Man-Month 2

ICS 121

- Program --> Program Product
  - Tested (esp. boundary values)
  - Documented (usage and maintenance)
- 3 X Cost of Simple Program
- Programming System --> Programming Systems Product
  - Precisely Defined [Module] Interfaces
  - Follows Prescribed Budget (system and organizational)
  - Tested (esp. its integration with other subsystems)
- 9 X Cost of Simple Program

## Myths

Topic 11  
Mythical Man-Month 3

ICS 121

- Poor Estimation
- Man-Month Myth
- Not Planning for Testing
- Gutless Estimating
- Regenerative Schedule Disaster

## Myths—Poor Estimation

Topic 11  
Mythical Man-Month 4

ICS 121

- Based on Assumption that Nothing Goes Wrong
- Large Project Consists of Many Smaller Tasks
- Probability of No Failures Diminishes

## Myths—Man-Month Myth

Topic 11  
Mythical Man-Month 5

ICS 121

- True: Project Cost is Proportional to Number of Personnel
- False: Progress is Proportional to Number of Personnel
- Fallacy is in an Assumption of Subtasks Requiring No Communication

## Myths—Not Planning for Testing

Topic 11  
Mythical Man-Month 6

ICS 121

- Many Projects on Schedule until Testing
- Bias toward No Failure
- Suggested Schedule
  - 33% for Planning
  - 17% for Implementation
  - 50% of time to Testing (half for component and half for integration)

# Topic 11 Mythical Man-Month

# ICS 121 Lecture Notes

Topic 11  
Mythical Man-Month 7

## Myths—Gutless Estimating

ICS 121

- Urgency of Client causes Optimistic Estimates
- Irregardless of Urgency, Tasks require same amount of time

Topic 11  
Mythical Man-Month 8

## Myths—Regenerative Schedule Disaster

ICS 121

- Adding Personnel Requires Retraining
- Retraining is not in the Planned Schedule
- Project Falls Further Behind
- Cycle Regenerates Itself

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

Topic 11  
Mythical Man-Month 9

## The Surgical Team (1)

ICS 121

- Surgeon
  - Expert Performs Design
- Copilot
  - Follows Design
  - Knows Alternatives

Topic 11  
Mythical Man-Month 10

## The Surgical Team (2)

ICS 121

- Administrator
  - Manages Money, People, Space, & Machines
  - Legal and Contractual Arrangements
  - Liaison between Surgeon and Client
- Editor
  - Reworks Surgeon's Documentation for General Consumption

Topic 11  
Mythical Man-Month 11

## The Surgical Team (3)

ICS 121

- Two Secretaries
  - for Administrator and Editor
- Program Clerk
  - Maintains Evolving Artifacts (versions etc.)
- Toolsmith
  - Expert on Supporting Software Tools
- Tester
  - Adversarial Role: Test Cases for Functional Tests
  - Assistant Role: Test Cases for Debugging
- Language Lawyer
  - Special Specification and Programming Language Features

Topic 11  
Mythical Man-Month 12

## The Surgical Team — Notes

ICS 121

- Based on 10:1 Ratio
- Scales up through Hierarchical Division of Problems
- Single Surgeon on Problem (Subproblem) maintains Conceptual Integrity of Design
- Requires Good Communication among Surgeons

# Topic 11 Mythical Man-Month

# ICS 121 Lecture Notes

Topic 11  
Mythical Man-Month 13

---

## Conceptual Integrity

ICS 121

- Conceptual Integrity = consistency and accuracy of model
- Conceptual Integrity implies Ease of Use
- Achieved More Easily with Fewer Designers (a surgeon/ architect-based approach)
- Achieved More Easily with Fewer Functions
- Ratio of Productivity Gain to Cost [of System and Training] usually Decreases with Increased Functionality

Topic 11  
Mythical Man-Month 14

---

## The Second System Effect

ICS 121

- Systems Evolve to Include Esoteric Features
- In so doing, they fail to anticipate paradigm shifts.
  - e.g., overlay systems dying before virtual memory systems
- Solution: Experienced System Architect Sensitive to the Second System Effect

Topic 11  
Mythical Man-Month 15

---

## Achieving Effective Communication

ICS 121

- Direct
  - Through Informal Mechanisms (e.g., telephone)
- Meetings
- Project Workbook
  - All Documents and Artifacts from Design through Implementation and Testing

Topic 11  
Mythical Man-Month 16

---

## Developer Productivity

ICS 121

- Not All Working Hours are Devoted to a Project
  - Interruptions include meetings, high-frequency, unrelated tasks.
- Productivity is constant in the units.
- Higher Level tools Imply Higher productivity

Topic 11  
Mythical Man-Month 17

---

## Pilot Systems

ICS 121

*Plan to throw one away; you will, anyhow.*

- Plan for Change — Mindset
- Plan for Change Organizationally
  - Training and Promotion Motivation

Topic 11  
Mythical Man-Month 18

---

## Pilot Systems Reconsidered

ICS 121

*Prototypes considered harmful!*

- Plan to throw one away?
- Or ...
  - Follow an incremental build strategy.
  - E.g., Microsoft Daily Build
  - E.g., Mockups and Scenarios
  - Not Prototypes

# ICS 121 Lecture Notes

Topic 11  
Mythical Man-Month 19

---

## Designing the Bugs Out (1)

ICS 121

- **Bug-proofing the Definition a.k.a. Conceptual Integrity**
- **Testing the Specification**
- **Top-down Design**
  - Allows Design by Single or Small Number of Architects
- **Structured Programming**
- **Component Debugging and Reuse**
- **Interactive Debugging**
- **Use Debugging Scaffolding**
  - as much as 50%

Topic 11  
Mythical Man-Month 20

---

## Designing the Bugs Out (2)

ICS 121

- **Control Changes**
  - Versioning
- **Add One Component at a Time**
- **Quantize Updates**
  - Large and Infrequent
  - or Small and Frequent

Topic 11  
Mythical Man-Month 21

---

## Documentation

ICS 121

- **End Users**
- **Acceptance Cases**
- **Modification [Flowchart]**
- **Self-Documenting Programs**

Topic 11  
Mythical Man-Month 22

---

## No Silver Bullet (NSB)

ICS 121

- **Essence**—the difficulties inherent in the nature of the software [complexity]
- **Accidents**—those difficulties that today attend its production but that are not inherent [production]
- **Solution: Grow Great Designers**

*p. vij, essence and accidents comes from commentary on Suzuki violin pedagogy*