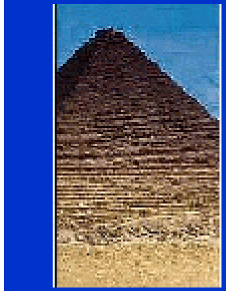


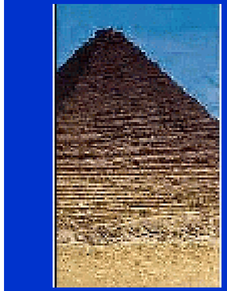
Pyrrhus Software
Enduring Solutions

The Soul of a New Machine

(revisited)

Joyce L Tokar, PhD

tokar@pyrrhusoft.com



Pyrrhus Software

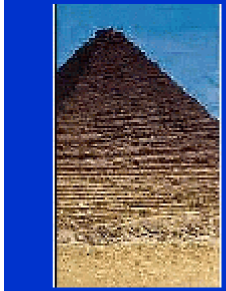
Enduring Solutions

Soul vs. Sole

Pyrrhus Software

Enduring Solutions

Sole of a New Machine



Pyrrhus Software

Enduring Solutions

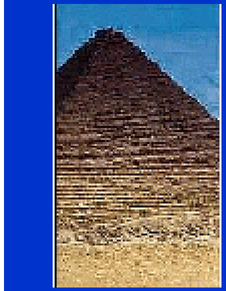
IEEE
How Russia Can Save the International Space Station p. 20
SPECTRUM
OCTOBER 2003
READY TO WARE
From Color-Changing Couture to Pulse-Sensing Bodysuits, Smart Garb Is Fashion That Works
PLUS
Avoiding Friendly Fire On the Battlefields Of Iraq p.42
The Top 100 R&D Spenders p.33
China to Microsoft: We'll Take Linux p.17
IEEE
DISPLAY UNTIL: 3 NOVEMBER 2003

The Soul of a New Machine

(Revisited)

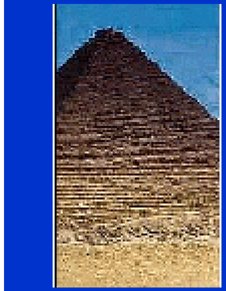
Tracy Kidder

1981 Avon Books
ISBN 0-380-59931-7

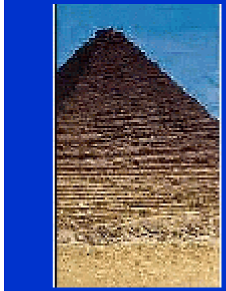


A Little Bit of History

- Late 1960’s, DEC is the leader in the Minicomputer market.
- 1968, Data General aggressively enters the Minicomputer market.
- 1978, Data General is a Fortune 500 company giving DEC a good run for their money in the Minicomputer domain.
- 1978, DEC releases the first 32-bit Minicomputer – the VAX.

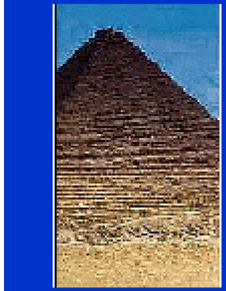


Once they are lost, both old and new customers are gone forever!



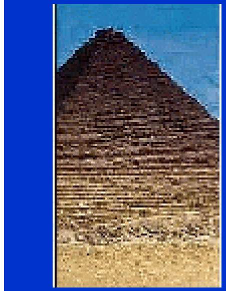
Innovations in Industry

- SPARK Ada & Raven SPARK
- Architecture Description Languages
- A#



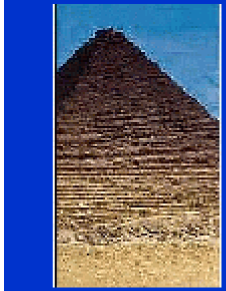
Software Compatibility

- Software is expensive.
- Getting software to function properly takes time.
- Software that works is precious.
- Software compatibility enables users to move to bigger and better machines without the large expense of rewriting the software.



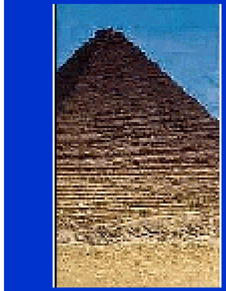
Software Compatibility

- Wrappers
- Emulators
- Virtual Machines
- Modeling



An Architecture Description Language (AADL)

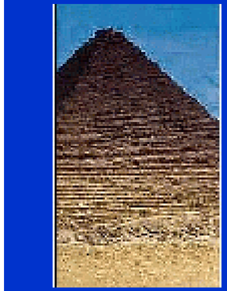
- At the highest level, AADL may be used to model an entire system – both hardware and software – independent of the final representation.
- Refinements to the model enable you to incorporate properties about both the hardware platform and the software components.
- Enables incremental development of systems composed of new components as well as legacy modules.
- Analysis of the model for schedulability and consistency throughout the development lifecycle.



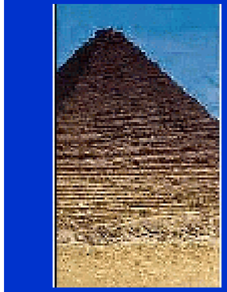
Let’s Show Them What We Can Do

Rules of the Game:

1. You must compete for resources.
2. Promise to achieve a nearly impossible schedule.
3. Get sign-up / commitment at all levels.
4. Success will generate reward.



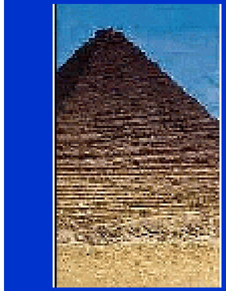
Not Everything Worth Doing Is Worth Doing Well.



Doing a Better Job

- Software engineering
- Spiral Development
- Patterns
- Collaborative Programming
- Extreme Programming

Avoid Over Engineering

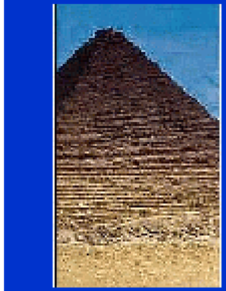


Building the System

- Define rules for the code and the hardware to fit together and to prevent each other from interfering with the other.
- Documented and used as the communication medium between the hardware team and the software team.
- Successful Hardware & Software developed in concert.

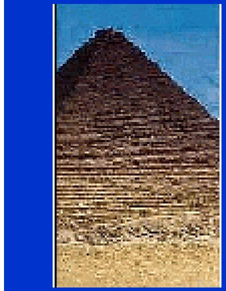
Communication Between Hardware & Software

- Programmer’s Interface Guide – PIG
- Architecture Description Language – AADL



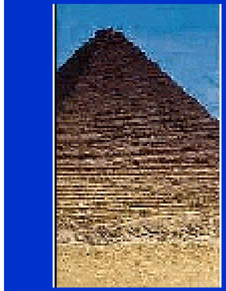
Hardware & Software Modeling with AADL

- From the hardware perspective AADL models include:
  - Processors
  - Memory
  - Devices
  - Buses
- From the software perspective AADL models include:
  - Threads
  - Processes
  - Systems
  - Packages
  - Subprograms
  - Data



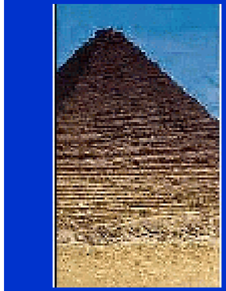
Analysis with AADL

- Can the system be built?
- Will it meet it's deadlines?
- Is it consistent?
- What is the impact of change?



Key to Success

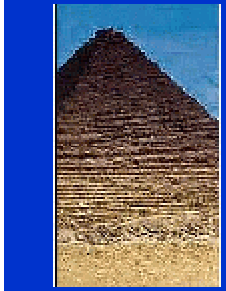
Engineers need the freedom to invent and the guidelines to success.



Pyrrhus Software
Enduring Solutions

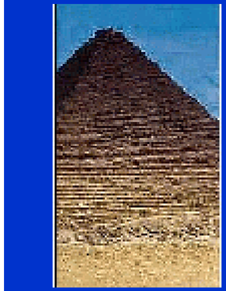
Innovation & Success

- CMM & CMMI
- Collaborative Programming
- Extreme Programming



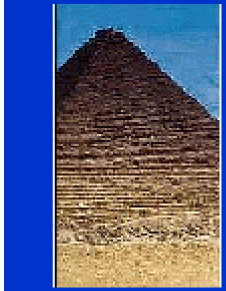
Software Costs

- Software Lifecycle
  - Requirements $1
  - Analysis $10
  - Design $100
  - Implementation $1000
  - Testing $10,000
  - Production $100,000
  - Deployed $1,000,000



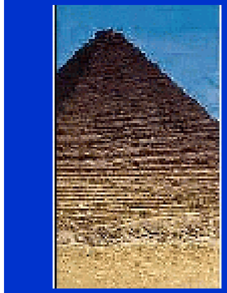
Extreme Programming

- Attitude –work as team on evolving the design. The customer is part of the team.
- Tests – define unit tests at the start of the project. Customer designs acceptance tests.
- Pair Programming – Two engineers per task produces more reliable code quickly.
- Refactoring – simplify the end product to remove redundancy and ambiguity.
- Incremental Growth – complete a set of agreed upon tasks then move on to the next set.
- Technology – tools the enable continuous testing and encourages continuous change.
- Communications – keep the customer in the loop throughout development.



Lessons Learned

- Innovation in Industry
- Software Compatibility
- Sign-Up & Commitment
- Good and Timely is better than Perfect and Late
- Hardware/Software Communication is Vital to Success Systems Development
- Encouraging & Guiding Engineering Teams



Thank You