CitySPIN Panel

December 9, 2003

UML and Software Engineering

Dr. Jorge L. Díaz-Herrera, Prof. & Dean

CitySPIN

The design of large, complex, mission critical applications has been in a *crisis mode* for many years and it is likely to remain so unless the community starts focusing on professional and standardization efforts.

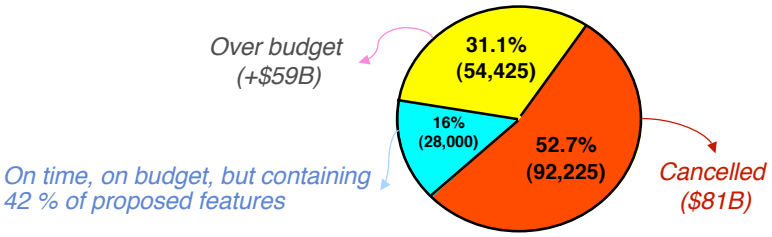
- Technology is not the problem
- Defined, managed standard processes are needed
- *Industrial-strength software engineering* must be part of the solution

*"(**UML™**) helps you specify, visualize, and document models of software systems, including their structure and design, in a way that meets all of these requirements."*
[www.omg.org]

Rising Cost of Software

"We have 500 projects. None are on time and on budget. This year, 40 percent will get canceled"

- US industry spends $250B/year IT application development for 175,000 projects*



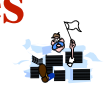
Jim Johnson, American Programmer 8, no. 7 (July 1995): 3-7

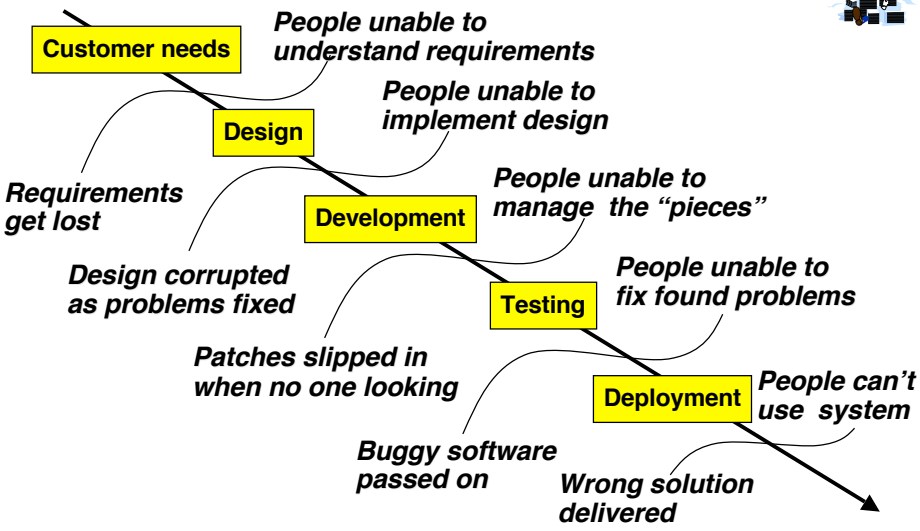
Related Causes

The problems are not essentially technical but the lack of a disciplined, engineering-based approach to use and manage the technology.

- few reliable data on software process; poor predictability among current practitioners; weak basis to evaluate new tools, methods etc.
- *inadequate formulation/understanding of requirements; poor communication between customer and developer*
- existing software can be hard to maintain ("legacy systems"); maintenance is typically more expensive than initial development

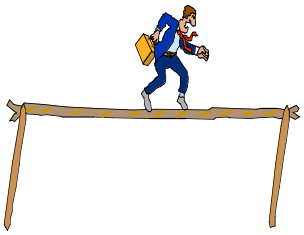
Sources of Difficulties





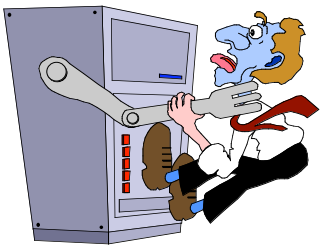
Software Difficulties*

Inherent Difficulties



Complexity
Changeability

Accidental Difficulties



Syntax
Representation
Technology

** Fred Brooks, "No Silver Bullet;" "Mythical Man-Month"*

Software *Engineering*

Engineering relies on **codifying** knowledge of ***commonly*** occurring problems in a form directly useful to ***practitioners***

- Adaptive design (**Software Architectures**)
  - *Use known, established solution principles and adapt the embodiment to changed requirements*
- Variant design (**Reusable Components**)
  - *Parts are varied within limits set by previously-designed product structures*

Modeling is the *design* of software before coding

- Capture design rationale and reasoning (decisions, paradigms)
- Facilitate team collaboration (common formalism) and communication (documentation)

UML & Software Engineering

UML is a *holistic* modeling language

- supports entire software development process
- Its 12 diagramming notations are “seamlessly” integrated

UML is a *general purpose* modeling language

- is applicable to different types of domain
- is fully extensible via “profiles”

UML is a method independent standard

- is an open language with wide industry/academia support
- XMI allows model interchange (regardless of tool/method)