Defense Program Lifecycle Management: A Dynamic Model for Policy Analysis

by

Jack B. Homer[1] and Ivan Somers[2]

[1]University of Southern California
[2]Hughes Aircraft Company

Defense Program Lifecycle Management: A Dynamic Model for Policy Analysis

ABSTRACT

The System Dynamics Lab at the University of Southern California's (USC) has worked with Hughes Aircraft Company's Electro-Optical and Data Systems Group to develop a system dynamics model for analyzing alternative policies available to a defense contractor for managing the production program lifecycle. Program lifecycle management is of prime importance to firms, like Hughes, that design, manufacture, and maintain complex military equipment. These firms have come under increasing government scrutiny and control, particularly with regard to cost and schedule risks.

The USC-Hughes model addresses the notion that cost and schedule risks can be substantially reduced through improved program management, even in the face of possible hurdles thrown up by customers and suppliers. The model suggests, for example, that overruns, particularly cost overruns, may be significantly reduced--without adversely affecting product quality--by carefully limiting the number and type of discretionary mid-production design improvements.

This presentation outlines the background and basic structural elements of the USC-Hughes model, demonstrates the model's ability to track historical data from two different cases, and highlights some of the policy findings that have emerged from the model.

Background

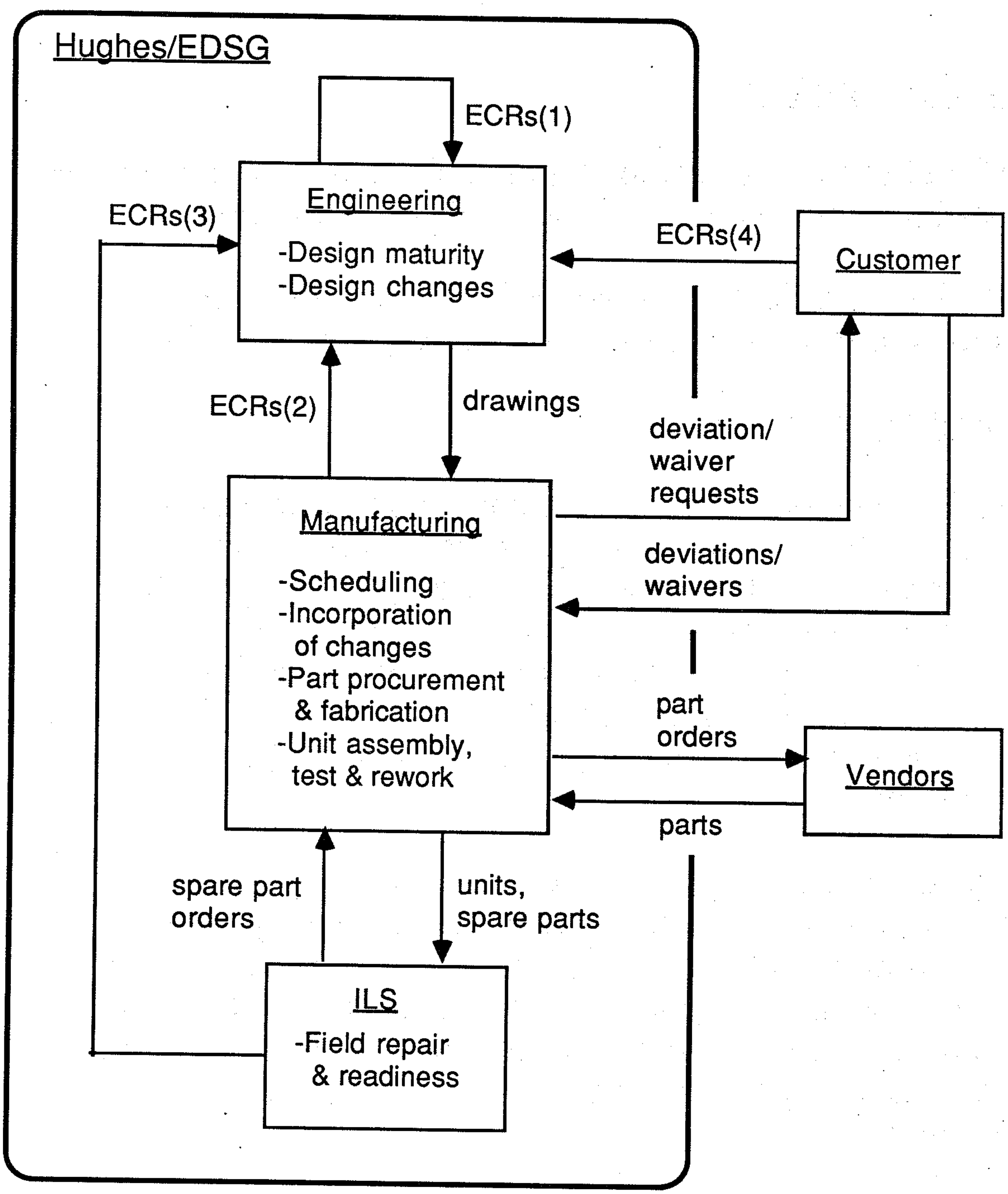
Client Motivation

- Project initiated in 1985 to analyze over-time impact of design changes on cost and schedule risk.
- Concerns over cost amplified in an environment of reduced Congressional budgets.
- Schedule slippages a chronic problem hurting customer relations and jeopardizing contracts.
- Overruns recognized as a dynamic problem: result of inefficiencies accumulating over entire program lifecycle of engineering design, manufacturing, and integrated logistics support (ILS).

Model Scope

- Model focuses on manufacturing and ILS phases of lifecycle. In particular, what happens when design imperfections are discovered mid-production?
- Significant design imperfections common in concurrent production programs: Government often requires manufacturing phase begin before design fully debugged and tested.
- Model focuses on flows of parts, assembled units, and engineering changes. Other factors (labor, equipment, cash, etc.) assumed available and non-constraining.

Overview of Model Structure



Sources of ECRs (Engineering Change Requests)

(1) Engineering analysis revealing design problems
(2) Test yield and producibility problems
(3) Field reliability problems
(4) New customer performance requirements

Case Studies and Policy Issues

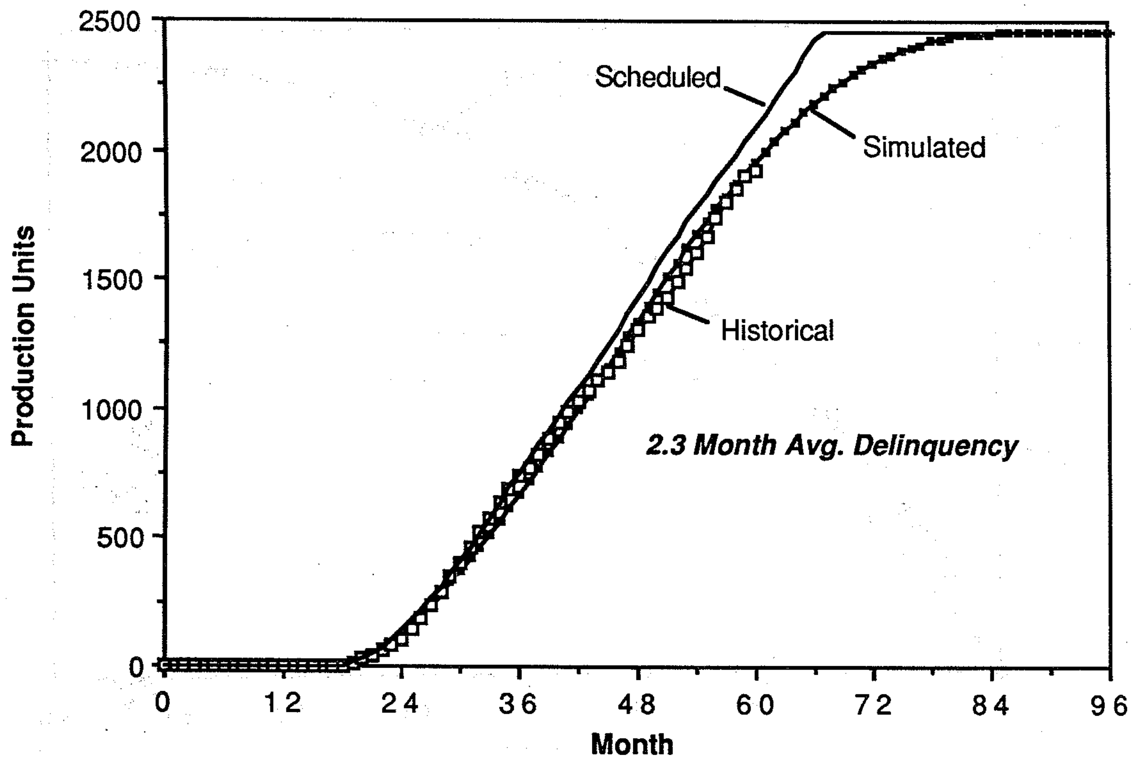
Two Case Studies

- Model initially developed and calibrated to represent one particularly troubled program still in production.
- Model "revalidation" involved recalibration to represent a different production program already completed.

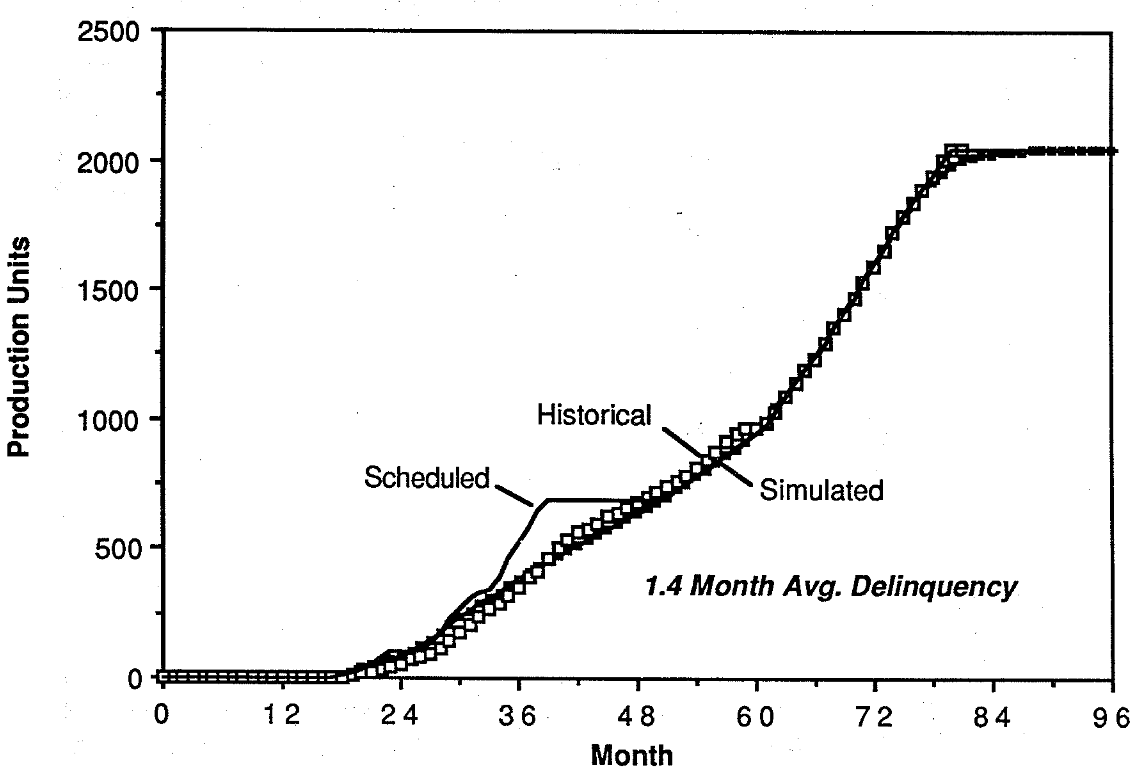
Central Policy Issues

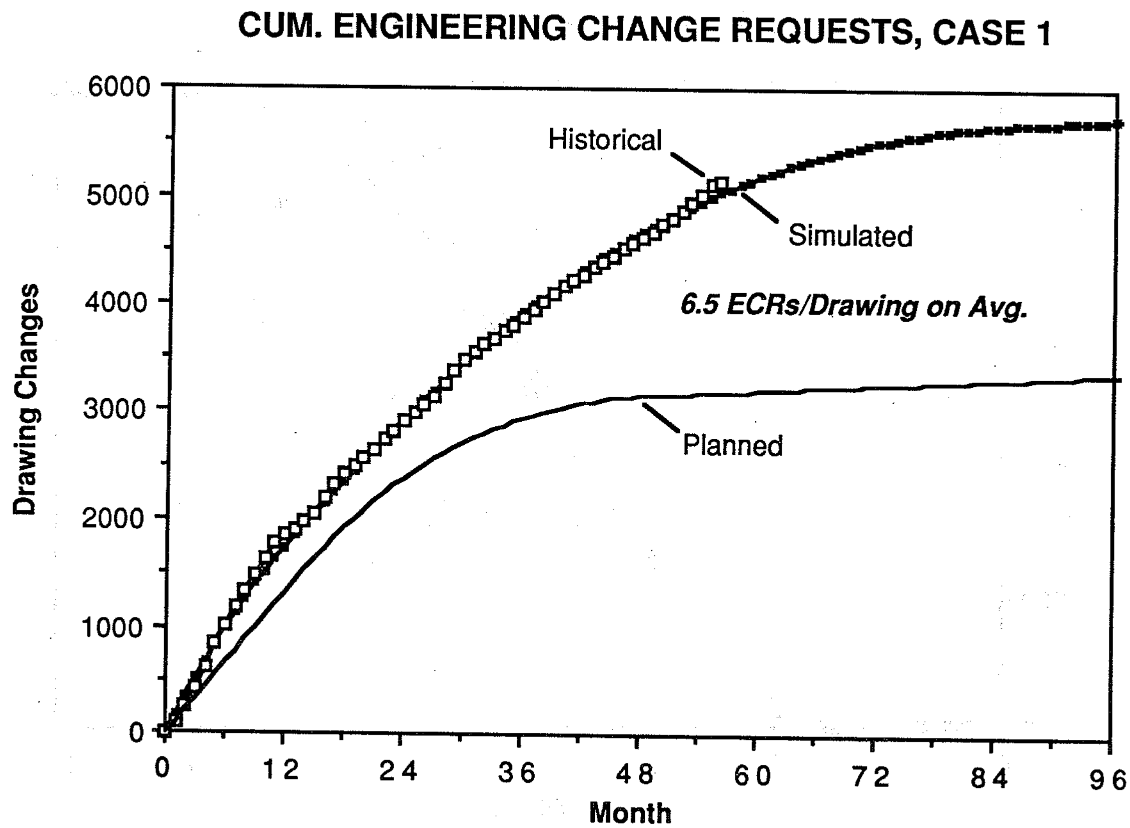
- Mid-production engineering change requests (ECRs):
  Most ECRs are internally generated rather than customer-directed. Should such discretionary ECRs be terminated altogether at some point during production? Should each be accepted only if its expected impact exceeds some minimum amount? Should they be grouped together and released in blocks rather than in a continuous stream?
- Disposition of old-version parts:
  When new-version parts arrive to the factory, it is a common practice to "purge" (discard) old-version parts from raw inventory and work-in-process. Indeed, not to do so may be considered a "deviation" from plans to roll new-version parts into production quickly. Should this practice of purging be modified?
- Ordering of parts:
  Because of uncertainty in vendor delivery times, it is a common practice to order parts so that most arrive well before they are needed for assembly. This is known as "front-loading" the parts delivery schedule. Should this practice be modified?

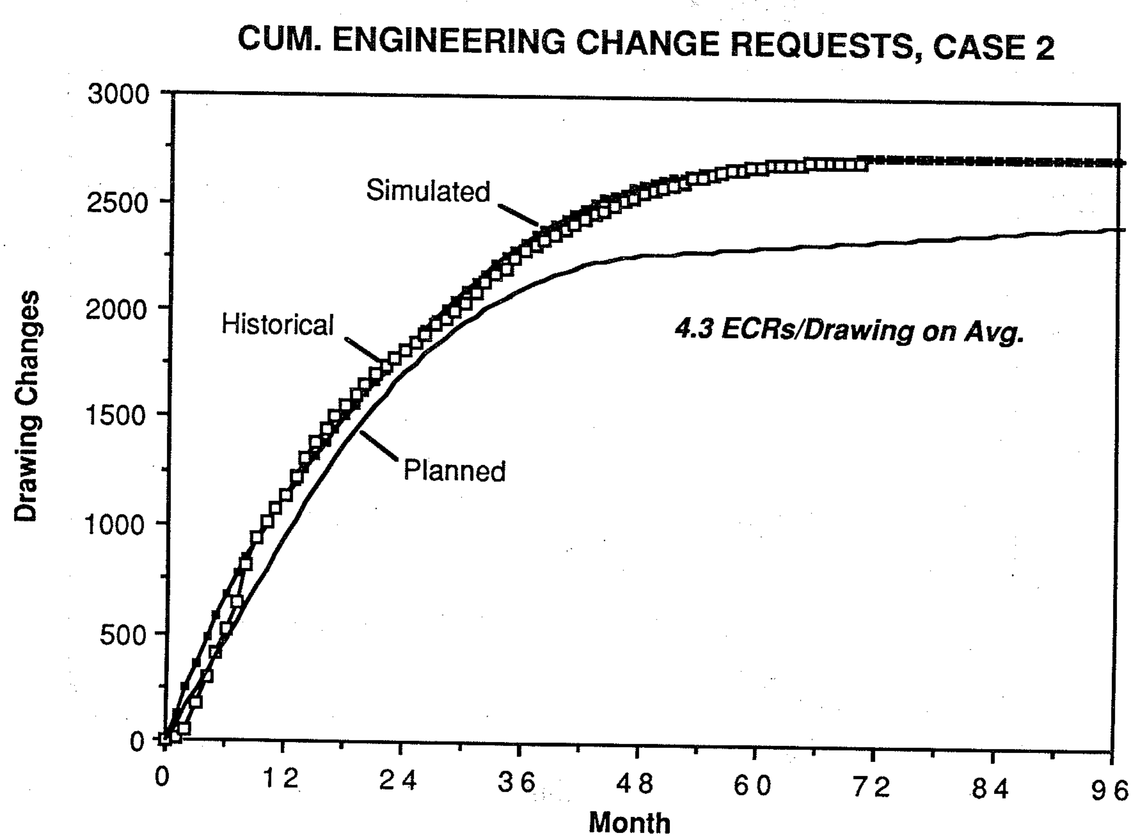
CUMULATIVE SHIPMENTS, CASE 1

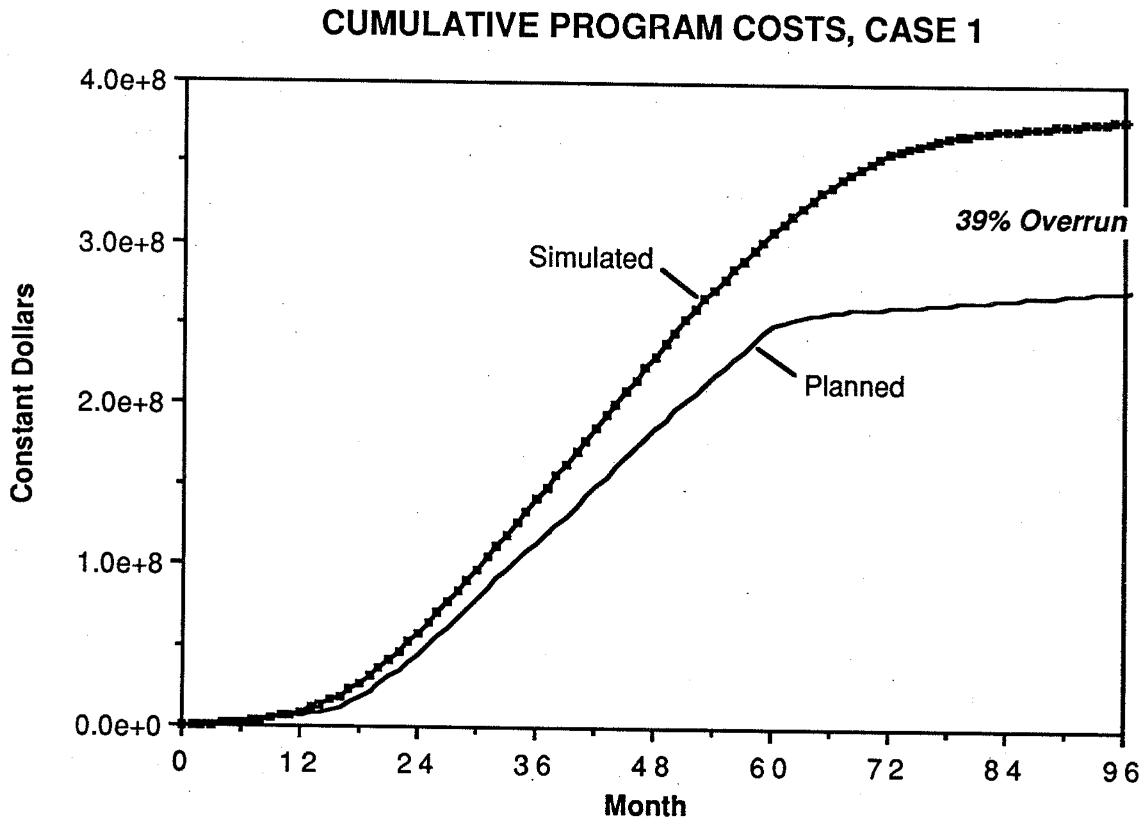


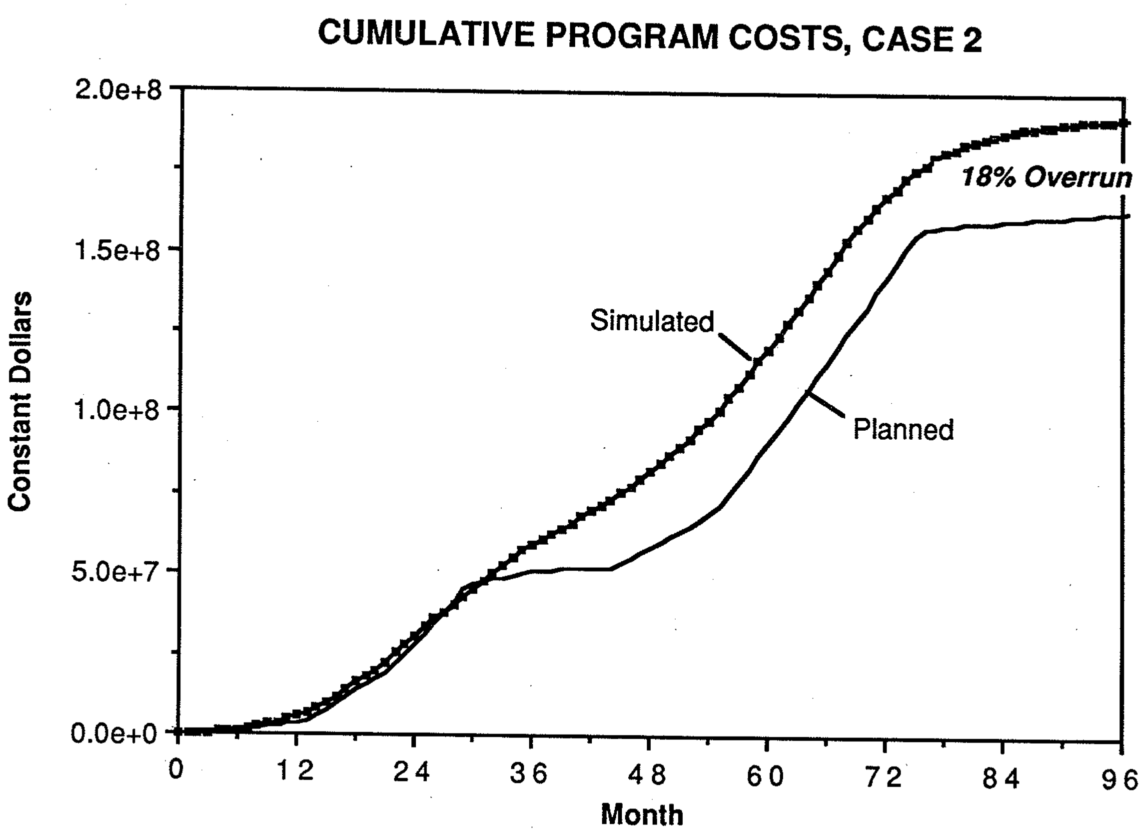
CUMULATIVE SHIPMENTS, CASE 2



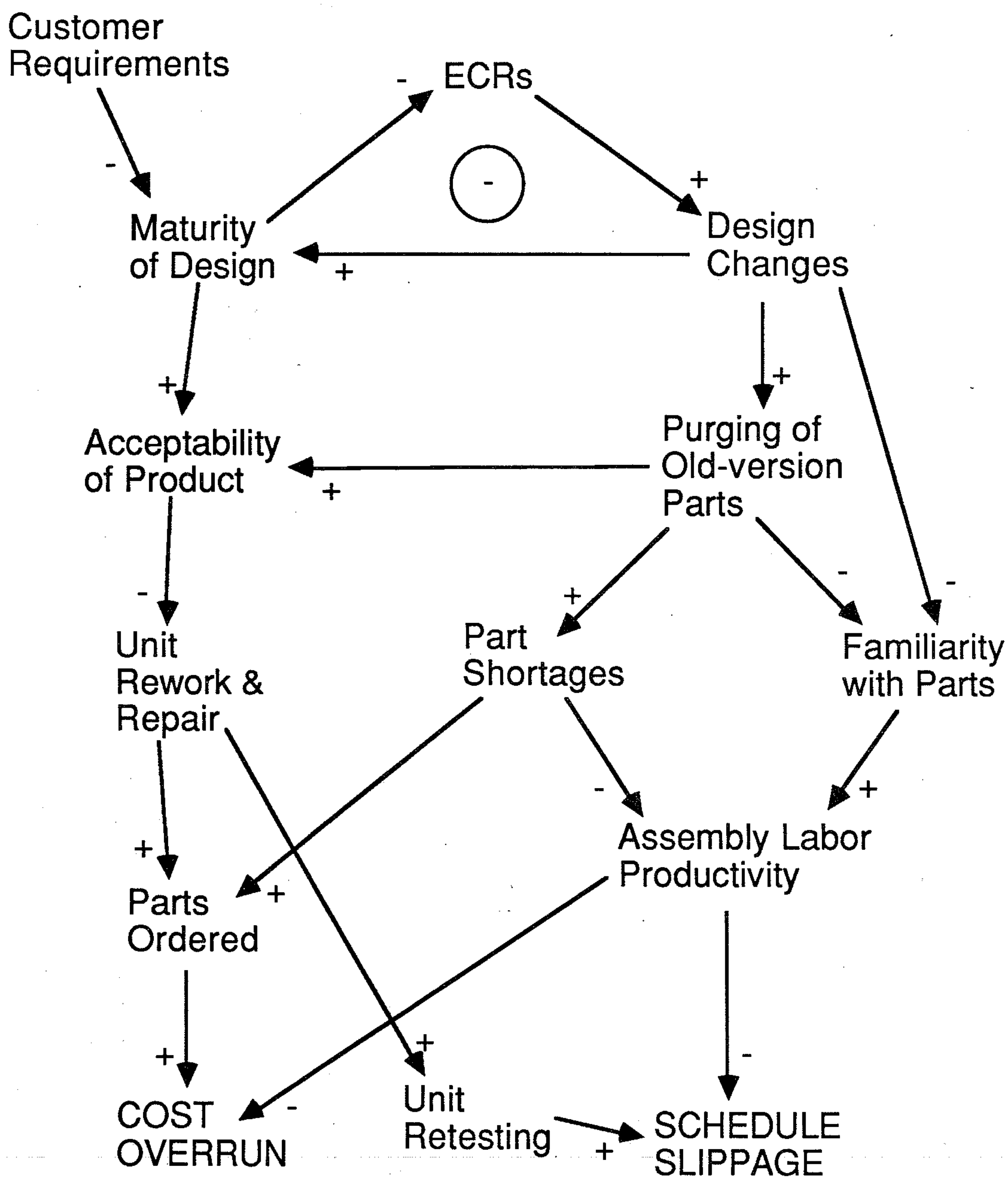

CUM. ENGINEERING CHANGE REQUESTS, CASE 1
Drawing Changes
6000
5000
4000
3000
2000
1000
0
Historical
Simulated
6.5 ECRs/Drawing on Avg.
Planned
0
12
24
36
48
60
72
84
96
Month


CUM. ENGINEERING CHANGE REQUESTS, CASE 2
Drawing Changes
3000
2500
2000
1500
1000
500
0
Simulated
Historical
4.3 ECRs/Drawing on Avg.
Planned
0
12
24
36
48
60
72
84
96
Month


CUMULATIVE PROGRAM COSTS, CASE 1
4.0e+8
3.0e+8
2.0e+8
1.0e+8
0.0e+0
Constant Dollars
0
12
24
36
48
60
72
84
96
Month
Simulated
Planned
39% Overrun


CUMULATIVE PROGRAM COSTS, CASE 2
2.0e+8
1.5e+8
1.0e+8
5.0e+7
0.0e+0
Constant Dollars
0
12
24
36
48
60
72
84
96
Month
Simulated
Planned
18% Overrun

Causal Structure Behind Problems of Cost and Schedule



Results and Recommendations

- Purging of old-version parts can generate severe parts shortages largely responsible for cost and schedule problems. It also disrupts assembly. Purging actually slows the rate at which acceptable units are produced, the opposite of its intended effect.

  Should eliminate part purging, except where part changes are customer-directed or "must-do" (i.e., non-discretionary). Reach understanding with customers on this policy so that use of old-version parts is not generally considered a deviation.

- Marginal benefit of ECRs decreases as design improves, but marginal cost remains the same. Also, marginal net benefit is greater early during manufacturing phase, before assembly comes up to speed. So, early termination of discretionary ECRs can be significant cost saver, also may reduce schedule slippage.

  Filtering-out of less significant ECRs followed by outright termination of all discretionary ECRs can cut costs more than termination alone, but the additional savings are relatively small, and appropriate implementation is tricky. The benefits of filtering may not be worth the extra effort and uncertainty involved.

  Block release of mid-production design changes delays their benefits and increases the disruption they cause. They should be released without blocking.

Results and Recommendations (continued)

- Front-loading of the parts delivery schedule builds up raw parts stocks and disguises the extra attrition caused by unexpected rework and repair (plus any purging). This can lead to the factory being caught short of parts at end-of-contract should the production program be temporarily or permanently discontinued. This may result in much additional schedule slippage. Should eliminate the practice of front-loading.

- When tested independently, "no part purging" and "ECR termination" policies each led to major theoretical reductions in cost and schedule overruns for the two historical cases.

  But "no part purging" had the greater impact, and combination of the two policies led to relatively minor additional cost improvement (and no additional schedule improvement) over "no part purging" alone. Indeed, the beneficial impacts of "ECR termination" taken alone are largely due to its indirect elimination of much purging.

- Whether tested independently or together with other policies, "no front-loading" policy led to sizeable theoretical reductions in schedule slippage for the two historical cases.