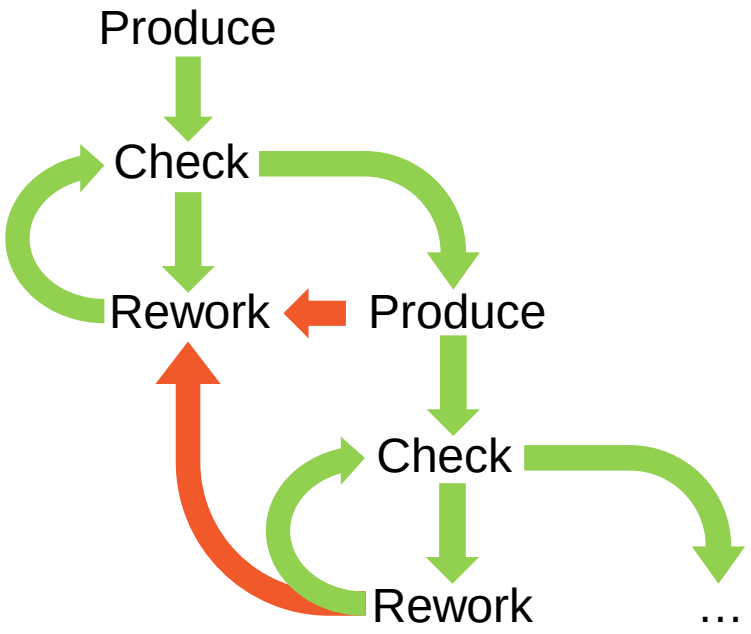
Test Scheduling Based on Rework Cycles

Dan Houston, Ph.D.

The Aerospace Corporation
9 November 2012

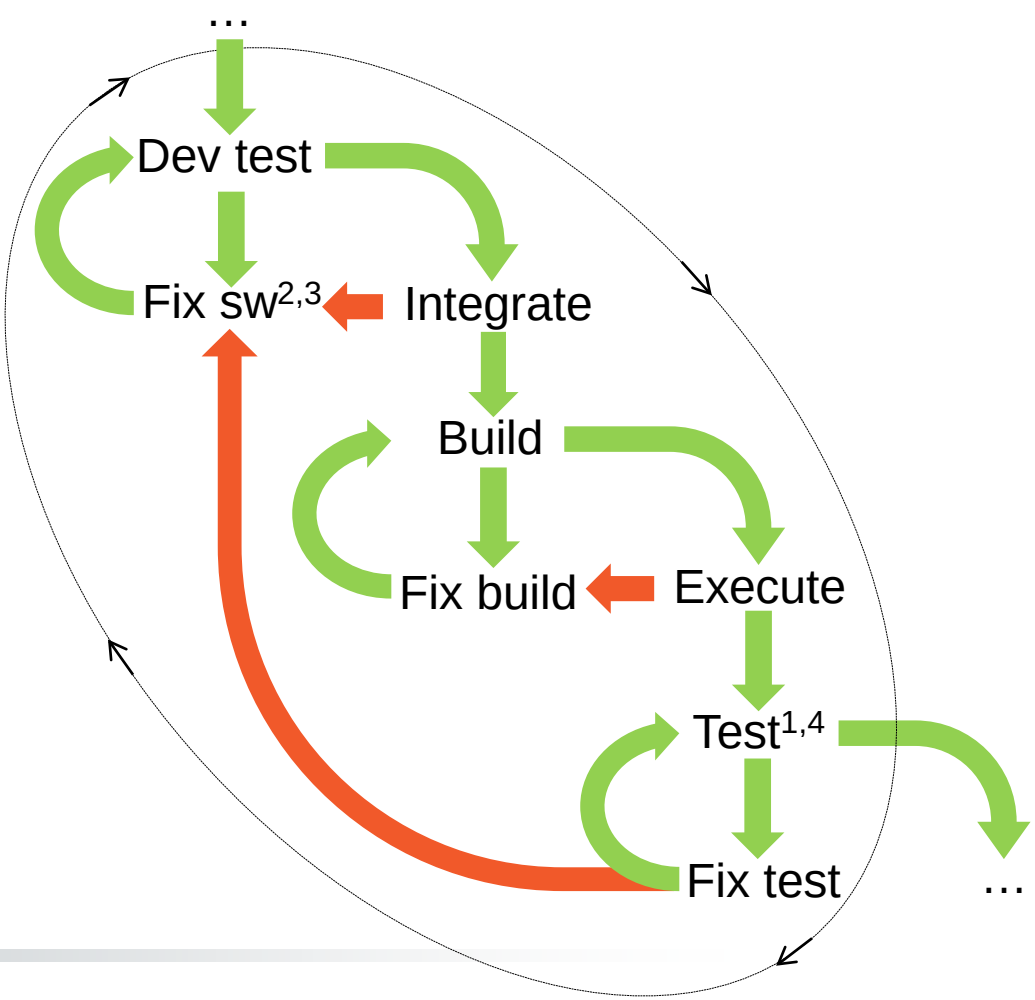
The Product Development Rework Cycle

- Studied extensively in the system dynamics modeling community
  - *Kenneth Cooper and the Ingalls case, 1980*
  - *Cooper's subsequent work*
    - Articles on the rework cycle
    - Consultancy on project change impact assessment
  - *Underlying problem: inability to account for quality and undiscovered rework in progress measures → quality shortcomings accrue until need for rework delays progress*
- Recognized as the central structure for modeling and simulating development projects
  - *Also called single most important feature, or canonical structure*
  - *Basis for much of the behavior in project models*



The Software Test-and-Fix Cycle

- The software test-and-fix cycle (TaFC) is a special case of the product development rework cycle
  - *It is especially challenging during integration testing of software that lacks sufficient effort spent on quality-inducing activities*
  - *Duration unknown*
- Common progress measures
  - *Test cases executed*[1]
  - *Defects found and fixed*[2]
  - *Defect backlog reduction*[3]
  - *Test cases completed*[4]
- Missing measure
  - *Number of rework cycles per test case*

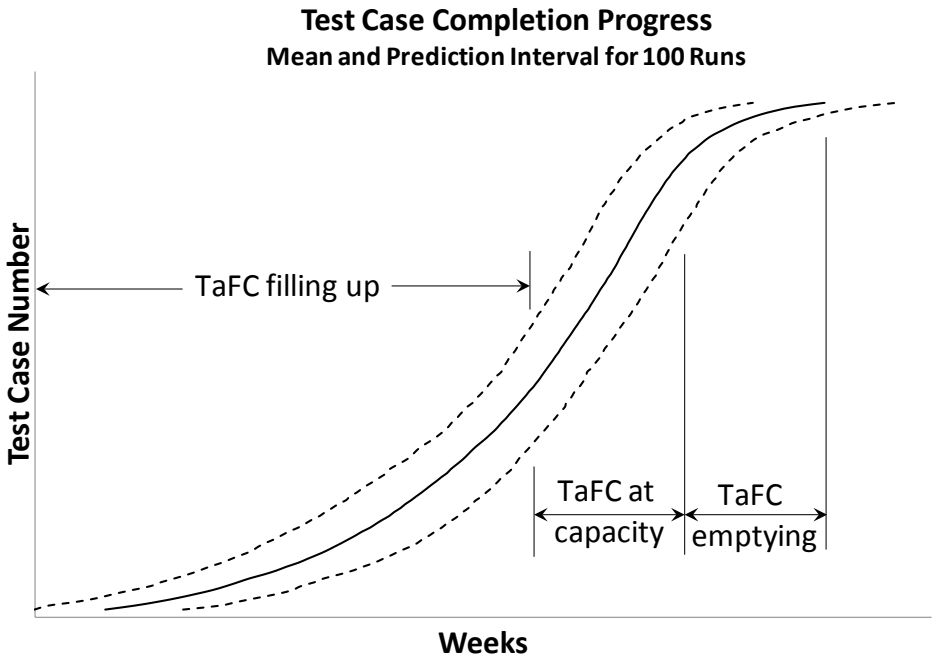


Preceding Work: Modeled Six Projects

- Same question for each, “How long will the software testing and fixing take?”
- Software
  - *Satellite systems (flight, payload or ground)*
  - *Produced by different organizations*
  - *Embedded and non-embedded, real time and non-real time*
  - *Development timeframes: several months to multiple years*
  - *Software characteristics, other than quality as reflected in rework cycles, was not a factor*
- Modeling focused on process characteristics
- Used discrete event modeling
  - *Process uses discrete measures, e.g., counts of test cases and rework cycles*
  - *Facilitates verification*
  - *Entities were test cases*
  - *Primary activities were testing and fixing*
  - *Primary resources were test facilities, testers, and fixers*

TaFC Duration Model Output

- Primary output: test case completions over time
- S-curve (assuming constant inputs for software availability and quality, test case availability, resource levels, and activities)
- Three stages
  - *Accelerating completions*
    - *Increasing slope*
    - *Process filling up with work*
    - *Duration is sensitive to WIP control*
  - *Steady completions*
    - *Constant slope*
    - *Process performing at capacity*
  - *Declining completions*
    - *Decreasing slope*
    - *Process emptying*



Forecasts and Results

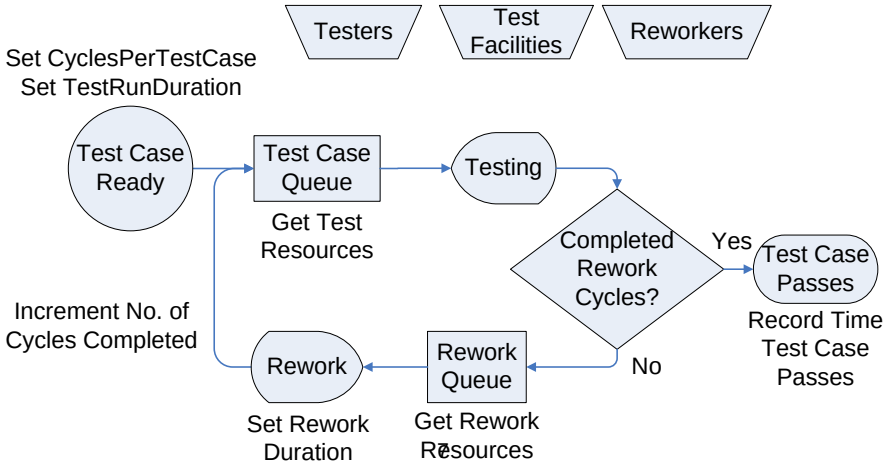
- Forecast:Plan Ratio is the ratio of the mean modeled duration (forecast) to the planned duration at the time modeling was performed
  - *Projects A, D, and F were large and the forecasting provided substantially better duration estimates*
  - *For smaller Projects B and E, the modeling confirmed existing schedules*
  - *Project C modeling provided a very early estimate of duration and resources but was not used to plan or track the test phase*

Project	Forecast:Plan Ratio	Actual within Prediction Interval?
A	2.5:1	Yes
B	1.0:1	Yes
C	n/a	n/a
D	2.0:1	9% high
E	1.0:1	Yes
F	1.2:1	Yes

- Although the first purpose of the six modeling projects was to answer the duration question, sensitivity analyses also provided information on dominant factors and supported process improvement decisions

An Analysis of Model Commonality and Variation

- Three dimensions
  - *Scope*
    - Variations include test case preparation, serial TaFCs, expert reviews, concurrent specialty testing
    - Focused on the core TaFC in each model
      - *Entities represent test cases and associated rework*
      - *Rework cycles per test case (not the same as number of defects)*
  - *Components*
    - Two bases for commonality
      - *Rework cycles contribute to duration*
      - *Test cases and associated rework are represented as TaFC entities*
    - A simple model of common components



AEROSPACE

Model Commonality and Variation

- Components (continued)
  - *Commonality*
    - Studied six models for common TaFC elements
  - *Variation*
    - Studied the model for variant elements
  - *Results of analysis (right box)*
    - Common elements in **bold font**
    - Variant elements in normal font
- Parameter values
  - *All vary, for example ranges of the simple model parameters*

Parameter	Range of Values
Number of test facilities	1 - 33
Number of testers	1 - 35
Number of software fixers	10 - 35
Number of test cases	30 - 2300
Number of rework cycles per test case	Distributions with means of 1.2 - 16
Test duration	Distributions with means of 0.1 - 9 hr
Rework duration	Distributions with means of 6 - 120 hr

1. Work shifts
2. Resources
   - **a. Test facilities**
     - **i. Number of test facilities**
     - ii. Availability of each test facility
       1. By shift
       2. Over days and weeks
   - b. Roles (for example, **testers**, analysts, **fixers**)
     - i. Number in each role (changing staff levels, for example, ramp-up and ramp-down)
     - ii. Shift(s) for each role
3. Workflow
   - a. Entities: Test cases and associated rework
     - **i. Number of test cases**
     - ii. Arrival time(s) of test cases
     - iii. Prioritization of test cases
     - iv. Levels of test cases
     - v. Work in process (WIP) control
     - **vi. Number of rework cycles**
   - b. Activities
     - i. Testing
       1. **Assignment and release of resources**
       2. Assignment of test case to test facility
       3. **Testing duration**
       4. Work shift limit on testing
     - ii. Anomaly analysis
       1. Assignment and release of resources
       2. Work shift limit on analysis
     - iii. Rework
       1. **Assignment and release of resources**
       2. Types of rework (**software fixes**, software patches, hardware, test tools, test scripts)
       3. **Rework duration**
          - a. Subcontractor interface delays
       4. Work shift limits on rework
     - iv. Software builds
       1. Adjustable build frequency

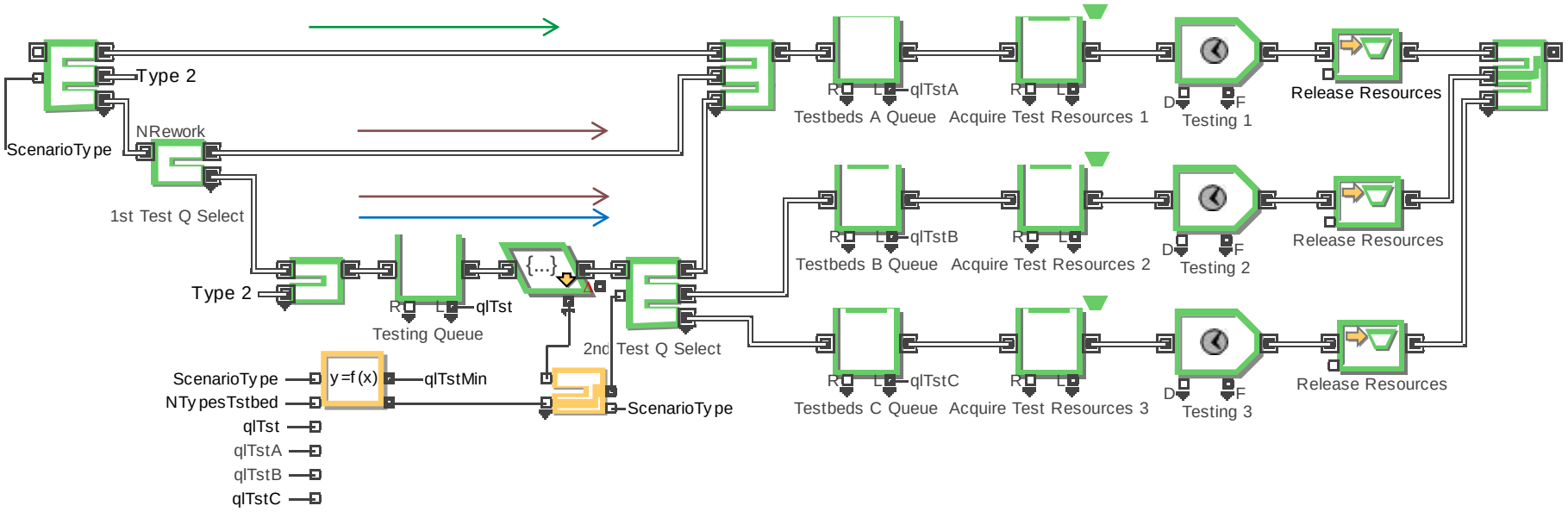
Types of Testing Scenarios Supported

- One type of testing
  - *All testing performed by one group of testers*
  - *One set of test cases*
  - *One type of test facility*
- Different types of testing
  - *Multiple types of testing, usually due to levels of integration*
  - *Multiple, concurrent TaFCs*
  - *Multiple sets of test cases, but sets may have many of the same test cases*
  - *Test facilities may differ by level of integration*
- Differently qualified test facilities
  - *One type of testing*
  - *Single set of test cases*
  - *Multiple test facilities distinguished by degree of operational emulation*
  - *Need to balance testing load across facilities*

A Generalized Test-and-Fix Model

- Testing workflow

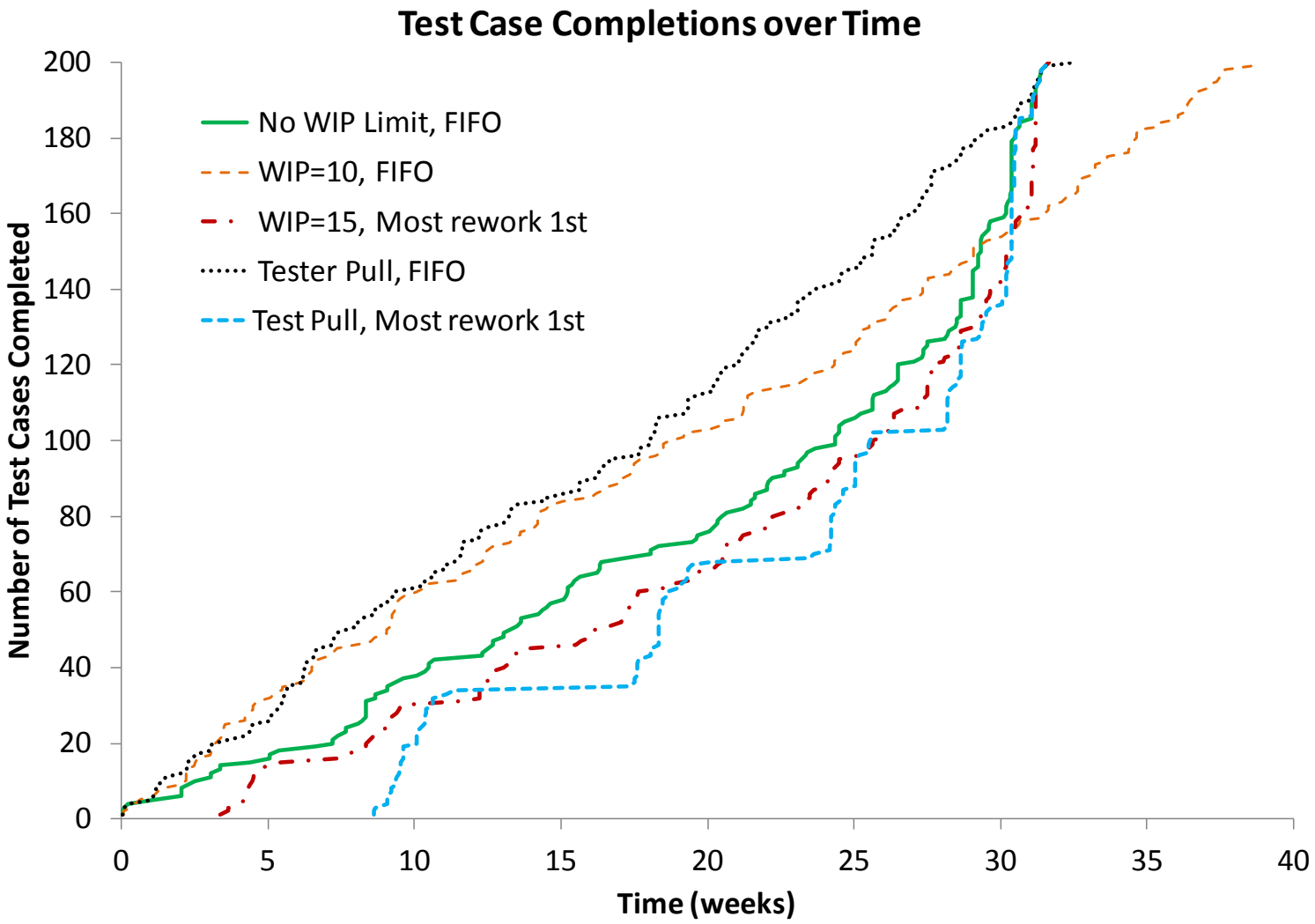
One type of testing Different types of testing Differently qualified test facilities



Implemented in ExtendSim

A Process Management Study

- Study two techniques used to manage workflow
  - *Work in progress (WIP)*
  - *Prioritization*
- Scenario type: all testing by 1 group using 1 set of test cases on 1 type of test bed
- Resources: 10 each of testers, testbeds, and software fixers
- Randomly distributed parameter values
  - *Test duration: Triangular(1, 2, 3) hrs*
  - *Rework duration: Triangular(4, 16, 40) hrs*
  - *Number of rework cycles per test case: UI(0, 6)*
- Four scenarios
  - *No WIP limit, FIFO*
  - *WIP = 10, FIFO*
  - *WIP = 15, most rework first*
  - *Tester pull, FIFO*
  - *Test Pull, most rework first*



Scenario	Max TIP (weeks)	Avg TIP (weeks)	Utilization	TaF Duration (weeks)
No WIP Limit, FIFO	31.7	21.0	0.97	31.7
WIP=10, FIFO	4.8	1.9	0.45	38.7
WIP=15, MR1st	5.4	2.3	0.56	31.7
Tester Pull, FIFO	11.7	5.1	0.92	32.5
Tester Pull, MR1st	11.3	5.2	0.97	31.7

Test Process Management Lessons

- A WIP limit reduces time in process and tends to “straighten” the test case completion line.
  - *However, as the WIP limit approaches the number of resources for the lowest capacity activity, it can constrain the work available for an activity, reducing utilization and causing an extension of the TaF process duration.*
- Prioritizing test cases by expected rework affects throughput
  - *Leaving hard test cases until late can prolong TaF duration*
  - *Doing hard test cases early slows initial TaF throughput*
  - *Ordering test cases by amount of expected rework, highest to lowest, reduces the likelihood of prolonging a TaF phase.*
  - *But in general, a moderate WIP limit or tester pull should be considered in conjunction with rework prioritization.*
- Low process capacity utilization can provide an opportunity for reducing process cost by balancing activity capacities.
  - *Seek to balance testing, analysis, and rework capacities.*
  - *One way of balancing the process is to allow testers to pull in test cases as needed and to staff rework to avoid a bottleneck.*

Conclusions and Further Work

- Analysis of the commonality and variation in six test-and-fix models produced to forecast duration of test phases
- Specified and implemented a generalized test-and-fix model
- Generalized model will be used to reduce modeling time of test-and-fix phases
- It will also provide a basis for further generalization
- Further studies
  - *Verify by reproducing results from each of the six predecessor models*
  - *Characterize process behavior over various regions of inputs*
    - Sensitivity analyses through designed experiments
    - Response surfaces for interesting regions and interactions
    - Possibly derive parametric formulae

Acknowledgment

This study was sponsored by The Aerospace Corporation's Software Acquisition Long Term Capability Development Project.