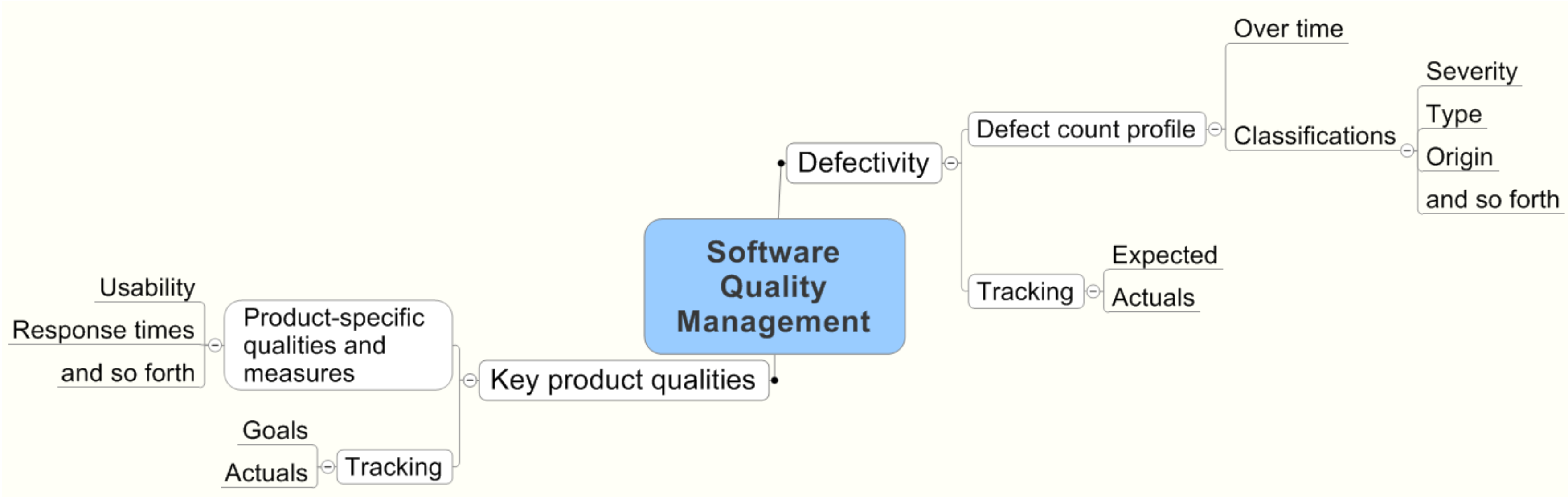
Dynamic COQUALMO:
Defect Profiling over Development Cycles

Dan Houston, Douglas Buettner, and Myron Hecht
The Aerospace Corporation

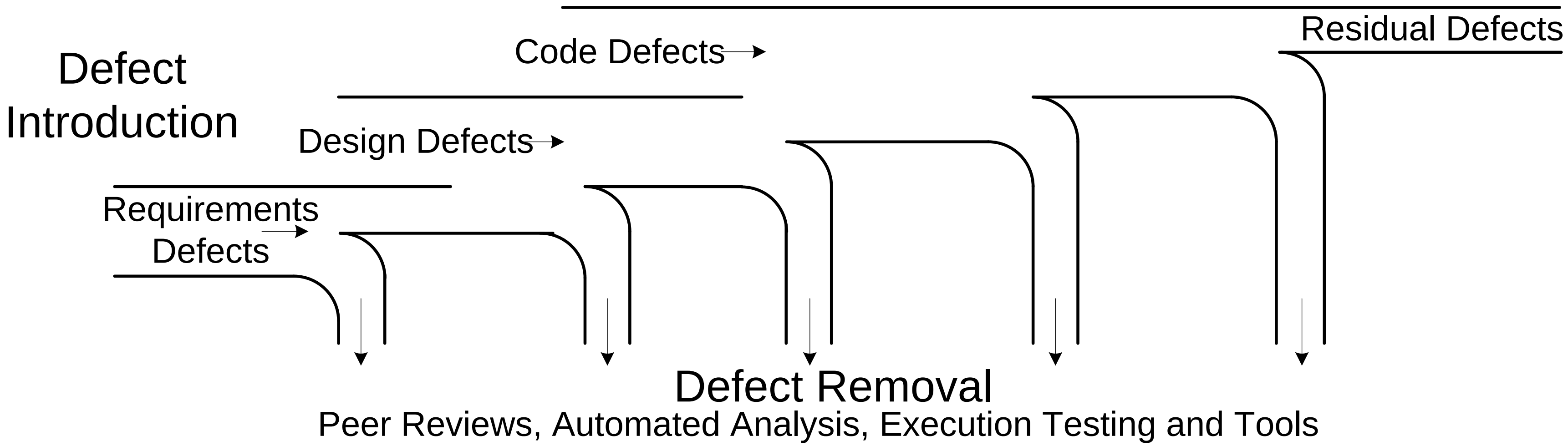
Computers and Software Division
May 16, 2009

Complementarity in Software Quality Management



- Product qualities
  - *Typically specific to a product or product line*
  - *Wide variety of metrics, for example, data display response time or usability*
  - *Analysis: regression, time series, designed experiments, simulation, etc.*
  - *Modeling supports proactive view of quality*
- Defectivity
  - *Generic approach and most common*
  - *Based on counts, categorization, and profiling over time*
  - *Analysis: defect profile targets, reliability growth, defect classification, leakage matrix*
  - *Most common derived metrics are defect density, defect discovery rate*

COQUALMO



- Extension of COCOMO II
  - *Relates defectivity to cost and schedule*
  - *COCOMO II drivers are treated as quality drivers*
  - *Quality measured in counts of non-trivial defects (critical system function impairment or worse)*
- Submodels
  - *Defect introduction*
  - *Defect removal*

COCOMO II and COQUALMO were developed at the Center for Systems and Software Engineering of the University of Southern California.

Defect Introduction Submodel

- Sources of defects: Requirements, Design, and Code

$$DI_{source} = DIR_{source,nom} * Size^{B_{source}} * \prod_{i=1}^{21} DefectDriver_{i,source}$$

- DI = defects introduced from each source
- DIR_{nom} = nominal defect introduction rate by source
- $Size^{B}$ = software size raised to scale factor by source
- Defect Drivers in Quality Adjustment Factors (QAFs)
  - *Example: Analyst Capability (ACAP)*
- Defect driver values produced through a two-round Delphi process.

ACAP Level	Requirements	Design	Coding
Very High	.75	.83	.90
High	.87	.91	.95
Nominal	1.0	1.0	1.0
Low	1.15	1.10	1.05
Very Low	1.33	1.22	1.11

Defect Removal Submodel

- Defect removal activities: peer reviews, automated analysis, testing

$$DR_{artifact} = DI_{artifact} * \prod_{i=1}^{3} (1 - DRF_{i,artifact})$$

- DR = defects removed from artifact
- DI = defects introduced into each artifact
- DRF = removal fraction for each activity, *i*, applied to each artifact
- DRF assigned to quality levels of activities in 2-round Delphi

AEROSPACE

Defect Removal Ratings

Rating	Peer Reviews	Automated Analysis	Execution Testing
Very Low	None	Simple compiler checking	None
Low	Ad hoc	Static module code analysis	Ad hoc
Nominal	Informal roles and procedures	Static code analysis; Requirements/design checking	Basic test process
High	Formal roles and procedures	Intermediate semantic analysis; Requirements/design checking	Organizational test process; Basic test coverage tools
Very High	Formality plus use of data	Temporal analysis & symbolic execution	Advanced test tools; Quantitative test process
Extra High	Review process improvement	Formal specification and verification	Highly advanced tools; Model-based test management

COQUALMO and Simulation Models

- In a model of software development project, add COQUALMO-based defect co-flows of artifact development
  - *Quality focus on residual defect density*
  - *Advantage: quality factors reflect dynamic project environment*
  - *Disadvantage: doesn't relate artifact defects to downstream activities*
  - *Choi and Bae (2006) developed a COCOMO II-based model*
  - *Tawileh et al. (2007) reused Abdel-Hamid and Madnick's model*
- Model only defect flows using COQUALMO, estimated durations, and Rayleigh curves
  - *Quality focus on defect management*
  - *Advantage: simulates dynamic project environment and defectivity profiling*
  - *Disadvantage: requires calibration with defect datasets*
  - *Madachy and Boehm (2008): Defectivity profile composed of generation and detection rates for each ODC type*
  - *This work: Defectivity profile composed of generation and detection rates for each activity*

Defectivity Profiling over Time

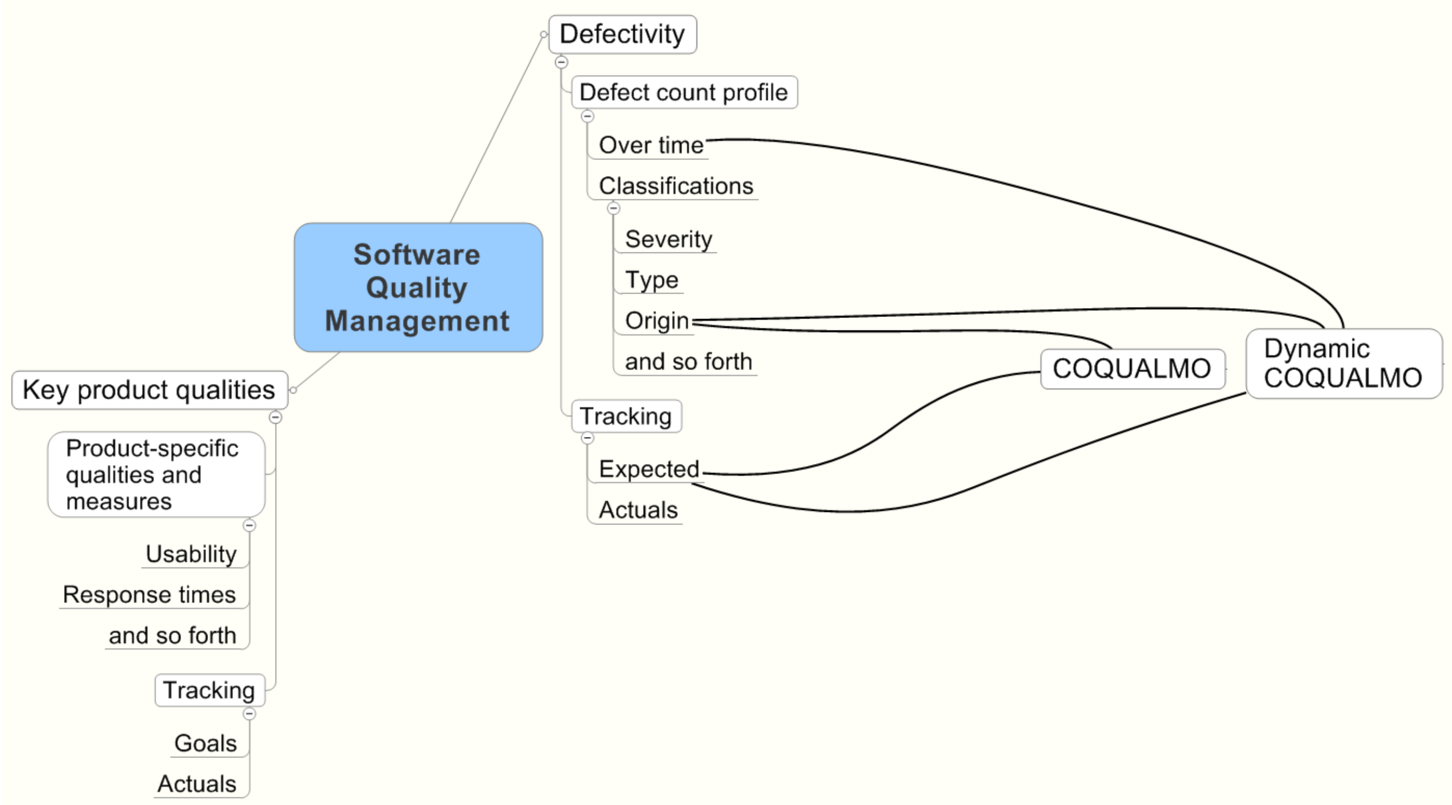
Fit distributions to defect discovery data

- In system test, reliability growth curves are used to estimate latent defects, support test decisions, support readiness for release decisions
- In earlier stages
  - *Reduce cost of software quality*
  - *Challenge: obtaining defect data*
  - *Answer: software inspections provide data for defectivity profiling*
  - *Example: defect leakage matrix*

		Defect Removal Phase								
		Requirements	Design	Code & Unit Test	SW Integration Test	System Test	Operational Test	Post-Release	Uncorrected	Leakage
Defect Injection Phase	Requirements	17	11	7	5	5	3	4	2	69%
	Design		71	56	21	11	6	3	2	58%
	Code & Unit Test			201	87	67	5	7	1	45%
	SW Integration Test				11	11	2	0	0	54%
	System Test					14	1	6	0	33%
	Operational Test						8	0	0	0%
	Post Release							0	0	
	Total New Defects	17	82	264	124	108	25	20	5	50%

Purpose of this Effort

- Advance defectivity profiling
  - *Utilize the quantitative relationships developed and refined in COCOMO II and COQUALMO.*
  - *Fit a non-parametric curve composed of multiple curves to defect data*



Model Description: Decompositions

- To accommodate significant project changes, provide for changes in defect driver and DRF values by project interval.
- To accommodate variation in quality of practice, use a profile (set of weighted values) for DRFs.
- To accommodate artifact types, use separate DRF profile for each artifact.
- To support reliability growth project, use two testing phases.

Defect Drivers by Artifact Type

- Project Intervals
  - Interval 1
  - Interval 2

Peer Reviews DRF

- DRF Profile
  - % Very Low
  - % Low
  - % Nominal
  - % High
  - % Very High
  - % Extra High
- Artifact Type
  - Requirements
  - Design
  - Code
- Project Intervals
  - Interval 1
  - Interval 2

Automated Analysis DRF

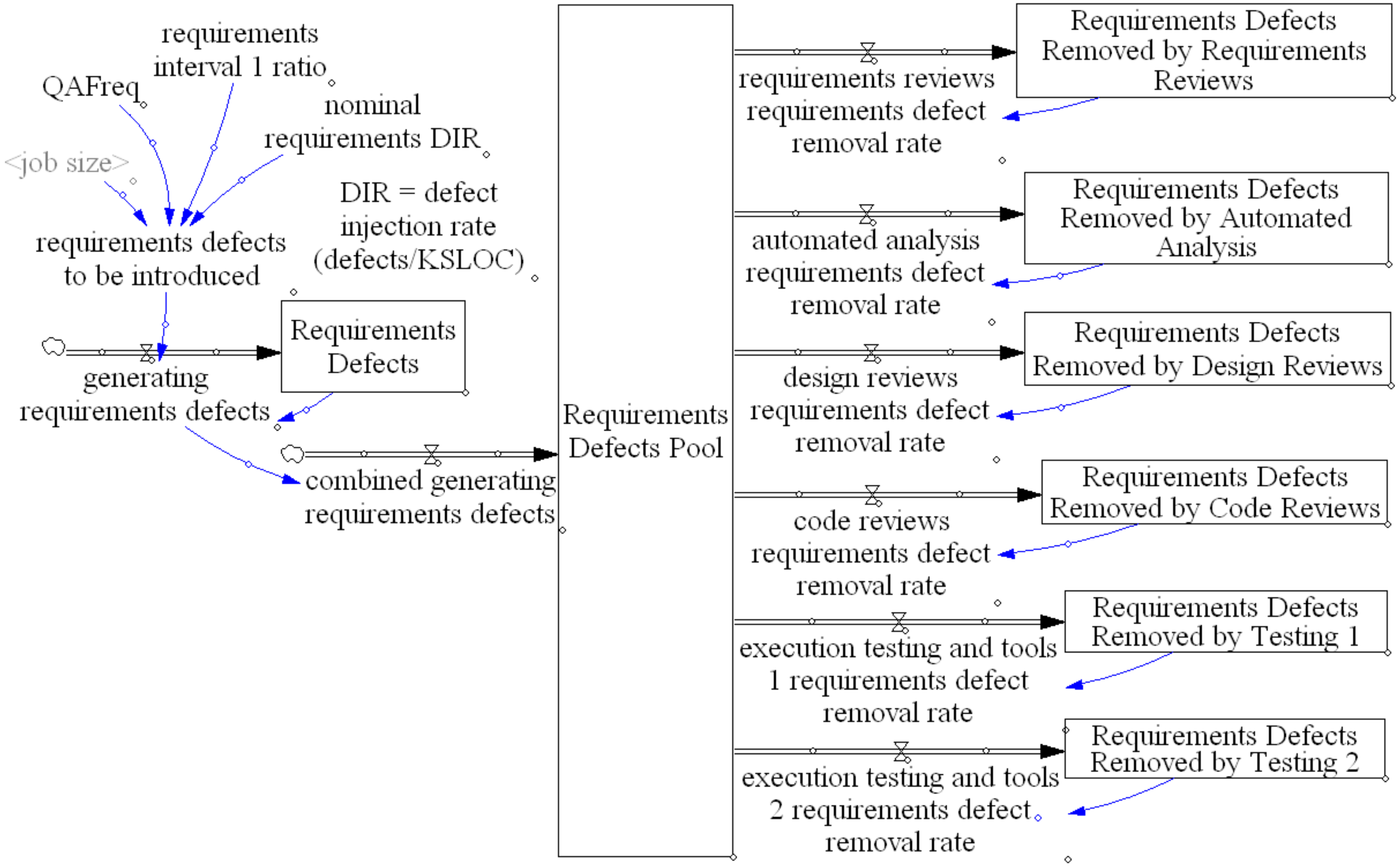
- DRF Profile
  - % Very Low
  - % Low
  - % Nominal
  - % High
  - % Very High
  - % Extra High
- Project Intervals
  - Interval 1
  - Interval 2

Testing DRF

- DRF Profile
  - % Very Low
  - % Low
  - % Nominal
  - % High
  - % Very High
  - % Extra High
- Testing Phase
  - Testing 1
  - Testing 2
- Project Intervals
  - Interval 1
  - Interval 2

Model Description: Defect Flows

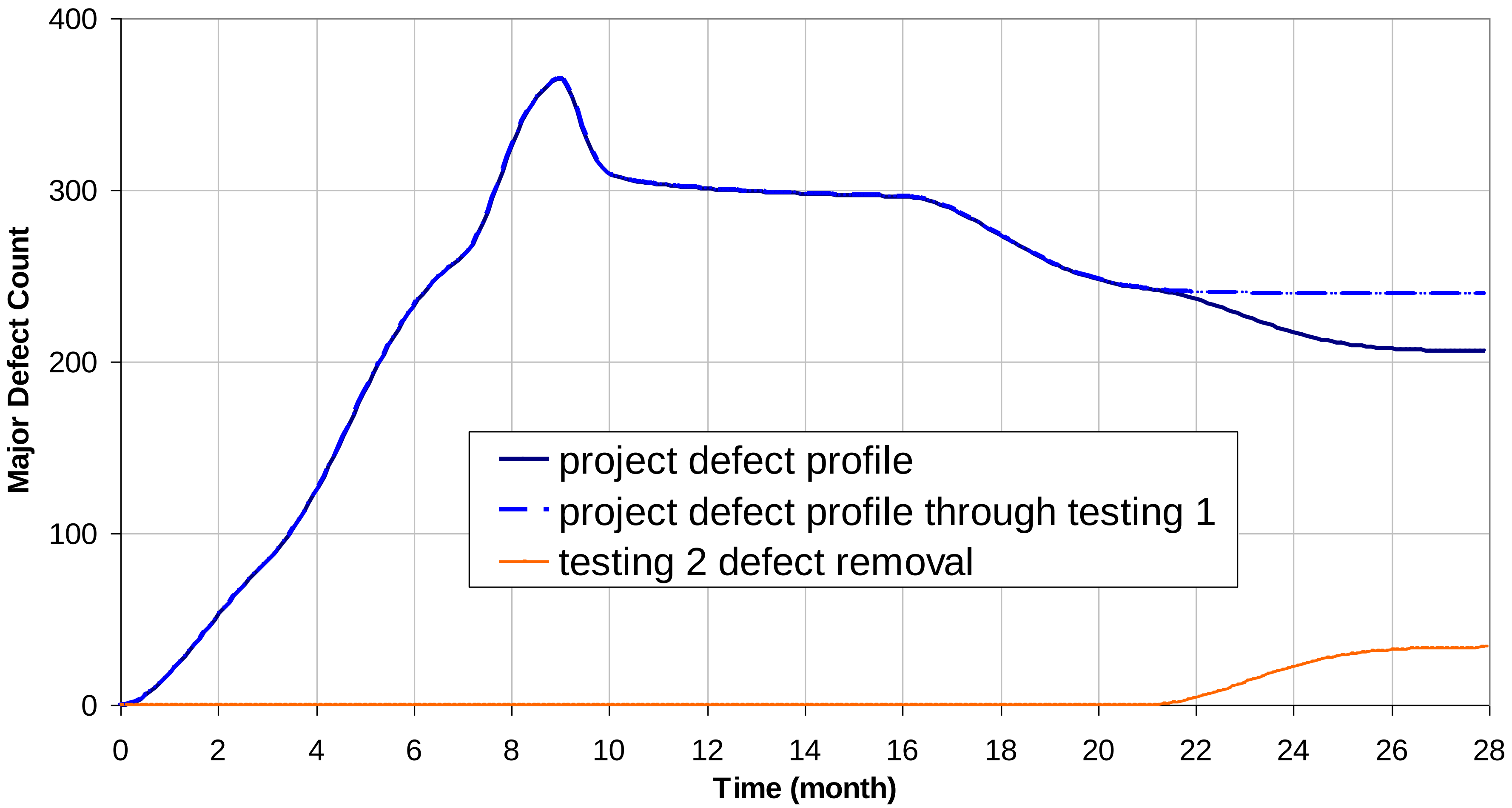
- Three inflows, one each for requirements, design, code
- Outflow for each review type, automated analysis, and testing phase
- Flows arrayed by interval



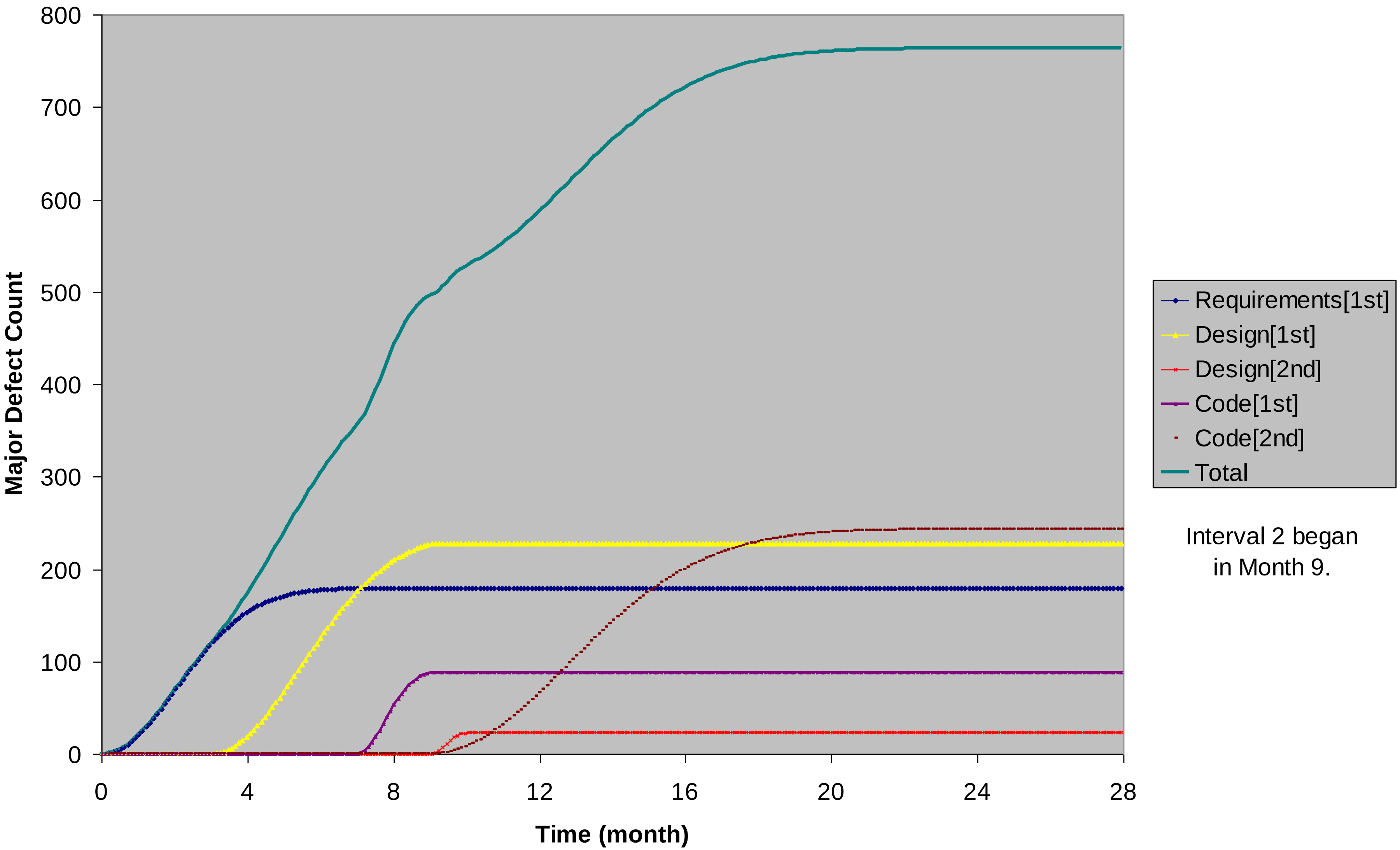
Model Description: Spreadsheet Inputs

- Estimated job size in KSLOC.
- Interval durations
- Estimated phase durations and degrees of phase concurrency such that they sum to the project duration.
- Delay from start of phase for starting peer reviews in each phase
- Relative effectiveness estimates:
  - *Relative effectiveness of requirements, design, and code reviews in finding requirements defects.*
  - *Relative effectiveness of design and code reviews in finding design defects.*
  - *Relative effectiveness of the two test phases in finding defects (requires definition of the differences between the two phases).*
- For each interval:
  - *Settings for defect drivers (COCOMO II factors), including effort multipliers and scale factors.*
  - *Usage profile of quality levels for each defect removal activity.*

Model Outputs: Project Defect Profile



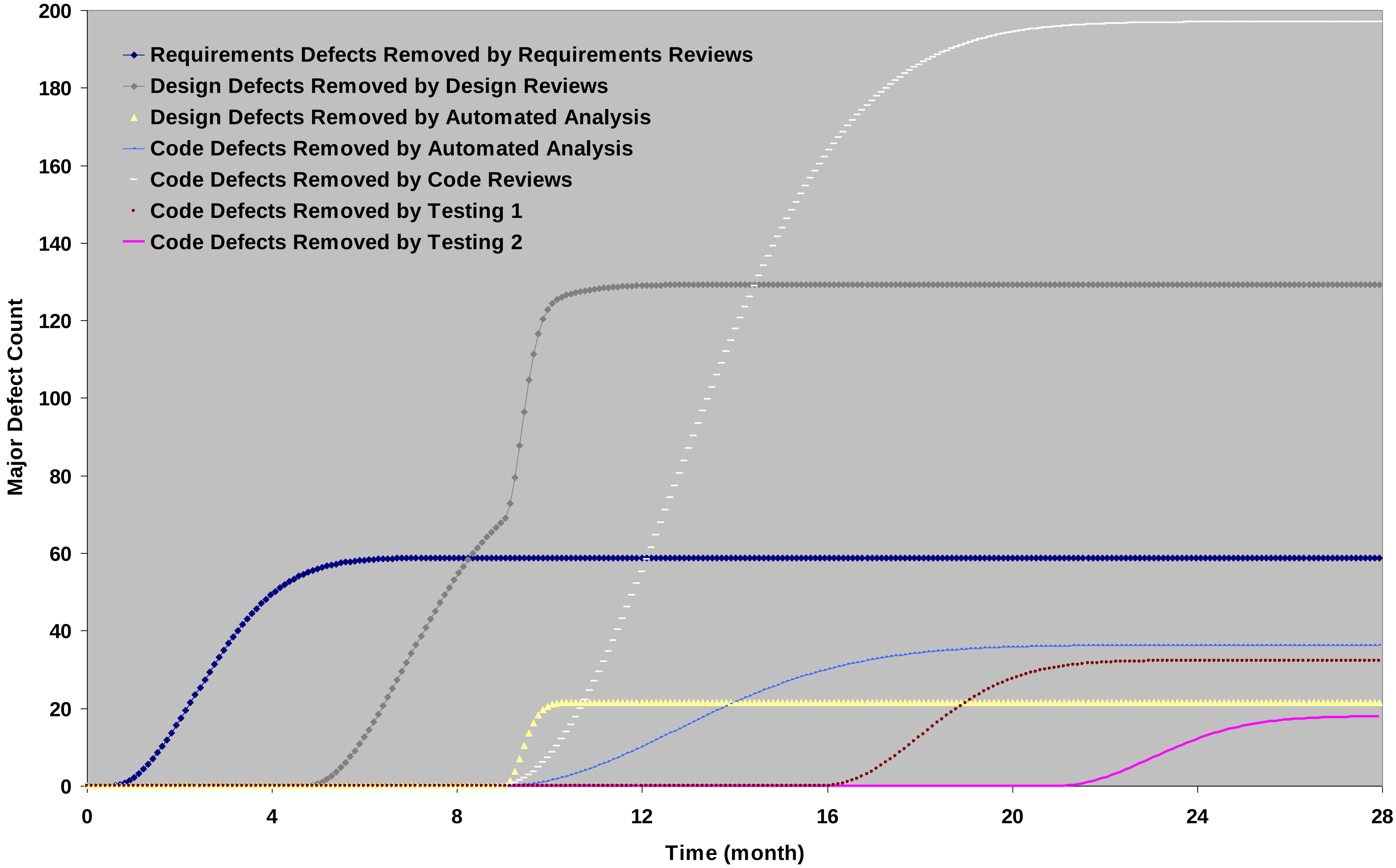
Model Outputs: Defects Introduced



Interval 2 began in Month 9.

AEROSPACE

Model Outputs: Defects Removed



AEROSPACE

Composing Defect Profiles with Rayleigh Curves

- Rayleigh distributions for project effort loading (Norden, Putnam)
- For a given set of project conditions,
  - *defect generation* $\propto$ *development effort*
  - *defect discovery* $\propto$ *defect generation*

  $\therefore$ use Rayleigh distributions represent defect discovery
- Project level: Trachtenberg (1982), Putnam and Myers (1995), Gaffney (1996), and Kan (2003)
- Phase level: Kan (2003), Modroiu and Schieferdecker (2006)
- Lower levels: Madachy and Boehm (2008)
- Activity level
  - *Intuitive appeal of shape*
  - *Easy to implement as function of amount flowing and time*
  - *Assumptions often satisfied "in the small"*

Rayleigh Curve Implementation

*rate = (total amount to be processed – amount processed) * time * buildup parameter*

- COQUALMO provides *total amount to be processed*
- Stock accumulates *amount processed*
- *buildup parameter = (coefficient * fractional duration ^{exponent}) / planned development duration*
  - *Generate Rayleigh curves for*

    3 months < planned development duration < 60 months

    .05 < fractional duration < 1.0
  - *Fit curves to results to obtain exponents and coefficients*
  - *Fit curves to exponents and to coefficents*

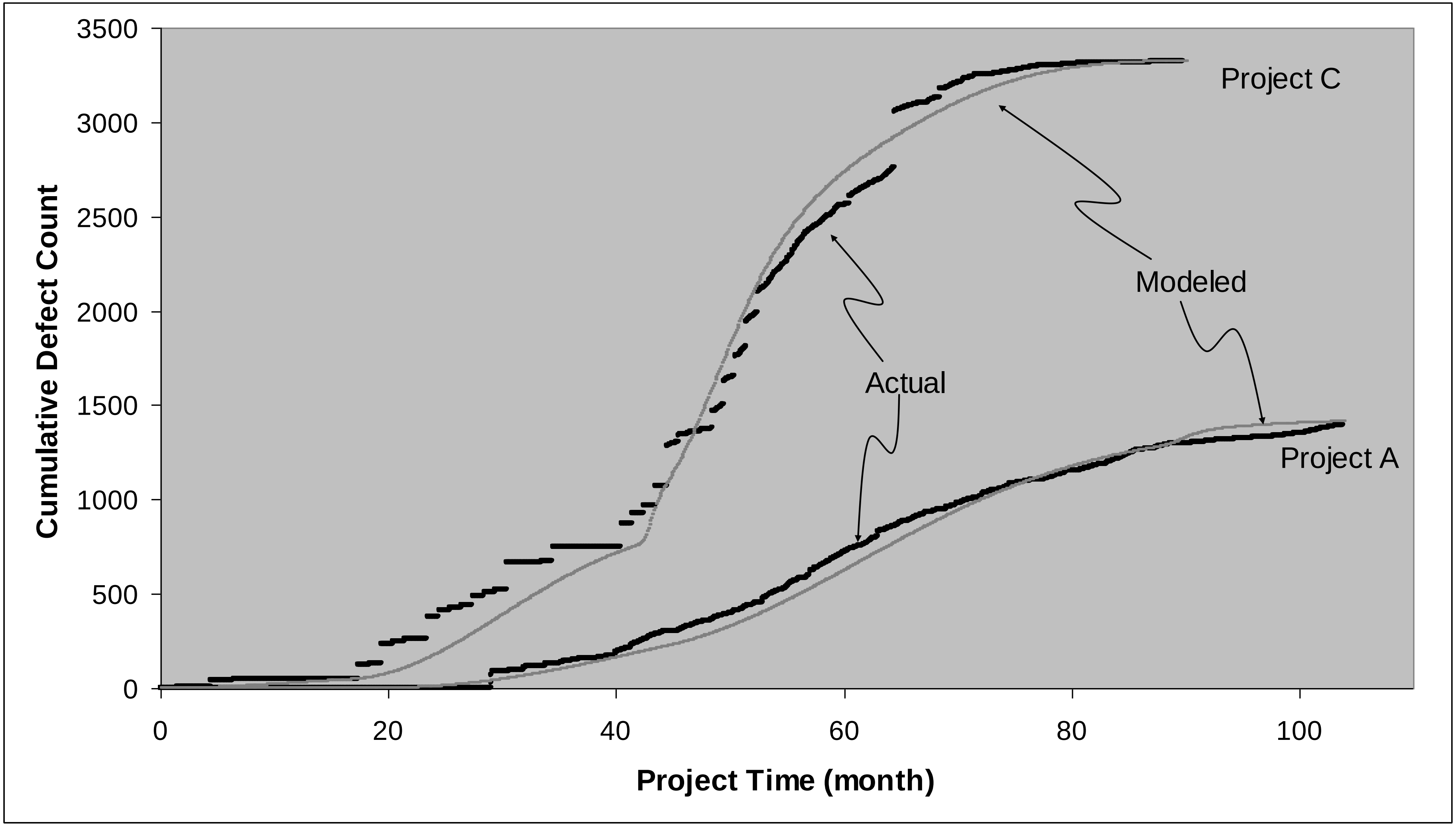
    *exponent = -0.01 * ln(planned development duration) - 2.0377* (R^2=.62)

    *coefficient = 6.3889 * planned development duration$^{(-1.0564)}$* (R^2=.99)

Model Testing and Usage

- Sensitivity
  - *Product size dominates*
  - *Next, nominal defect introduction values*
  - *QAFs (lognormally distributed)*
- Replication
  - *Two space system flight software projects*
  - *Project A: 68 KSLOC (Ada)*
    - Revised in its 8th year during testing.
    - Average QAF change from 3.2 to 1.5
  - *Project C: 99 KSLOC (50 Ada and 49 assembly)*
    - Redesigned during its third year.
    - Average QAF change from 11.4 to .31
    - Had better use of peer reviews, matured sooner.

Major Defect Discovery Profiles for Projects A & C, actual and modeled



Lessons Learned

- COQUALMO values for nominal defects introduced (10, 20, and 30 defects /KSLOC for requirements, design, and code) appear to be high.
  - *Values between .5 (Project C requirements) and 6.1 (Project C code) were used to produce the modeled curves.*
- The need to adjust the usage profiles suggests that either COQUALMO's DRF values require adjustment, or the usage of defect removal activities was reported inaccurately, or both.
- Software development projects seem to have characteristic defect discovery profiles.
  - *Dynamic COQUALMO can replicate a discovery profile and, by inference, produce a realistic defect profile for use in managing quality effort in an organization's future projects.*

Defect Removal Submodel

- Defect Drivers in COQUALMO QAFs
  - *Required Software Reliability (RELY)*
  - *Required Reusability (RUSE)*
  - *Process Maturity (PMAT)*
  - *Main Storage Constraint (STOR)*
  - *Analyst Capability (ACAP)*
  - *Applications Experience (AEXP)*
  - *Language and Tool Experience (LTEX)*
  - *Use of Software Tools (TOOL)*
  - *Development Schedule (SCED)*
  - *Precedentedness (PREC)*
  - *Team Cohesion (TEAM)*
  - *Data Base Size (DATA)*
  - *Product Complexity (CPLX)*
  - *Execution Time Constraint (TIME)*
  - *Platform Volatility (PVOL)*
  - *Programmer Capability (PCAP)*
  - *Platform Experience (PEXP)*
  - *Personnel Continuity (PCON)*
  - *Multisite Development (SITE)*
  - *Disciplined Methods (DISC)*
  - *Development Flexibility (FLEX)*
  - *Architecture/Risk Resolution (RESL)*
  - *Documentation Match to Life-Cycle Needs (DOCU)*

COCOMO II and COQUALMO were developed at the Center for Systems and Software Engineering of the University of Southern California.

AEROSPACE

References

- Devnani-Chulani, S.: Modeling Software Defect Introduction and Removal: COQUALMO (COnstructive QUALity MOdel). Technical Report USC-CSE-99-510, University of Southern California. (1999) http://sunset.usc.edu/publications/TECHRPTS/1999/ usccse99-510/usccse99-510.pdf Accessed July 23, 2008.
- Devnani-Chulani, S.: Bayesian Analysis of Software Cost and Quality Models. Doctoral Dissertation, University of Southern California, (May 1999)
- Buettner, D.J.: Designing an Optimal Software Intensive System Acquisition: A Game Theoretic Approach. Doctoral Dissertation, University of Southern California (2008)
- Choi, K.S., Bae, D.H.: COCOMO II-based dynamic software process simulation modeling method. Technical report CS-TR-2006-261, Computer Science Department, Korea Advanced Institute of Science and Technology, Daejeon, Korea. (2006) http://cs.kaist.ac.kr/research/technical/Archive/CS-TR-2006-261.pdf Accessed July 24, 2008.
- Abdel-Hamid, T.K., Madnick, S.E.: Software Project Dynamics. Prentice Hall, Englewoord Cliffs, New Jersey (1991)
- Tawileh, A., McIntosh, S., Work, B., Ivins, W.: The Dynamics of Software Testing. In Proceedings of the 25th System Dynamics Conference, July 29- August 2, 2007, MIT, Boston. http://systemdynamics.org/conferences/2007/proceed/papers/ TAWIL320.pdf Accessed October 7, 2008.
- Madachy, R., Boehm, B.: Assessing Quality Processes with ODC COQUALMO. In Wang, Q., Pfahl, D., Raffo, D. (eds.) Making Globally Distributed Software Development a Success Story, pp. 198-209 Springer-Verlag, Berlin (2008). R. Madachy also discusses the ODC COQUALMO model in Software Process Dynamics, Wiley & Sons, Hoboken, New Jersey (2008).
- Trachtenberg, M.: Discovering how to ensure software reliability. RCA Engineer (Jan-Feb 1982) 53-57.
- Putnam, L.H., Myers, W.: Familiar Metric Management—Reliability. (1995). http://www.qsm.com/ fmm_03.pdf Accessed July 21, 2008.
- Gaffney, J.: Some Models for Software Defect Analysis. Lockheed Martin Software Engineering Workshop, Gaithersburg, Maryland (Nov 1996).
- Kan, S.H.: Models and Metrics in Software Quality Engineering, 2nd ed. Addison-Wesley, New York (2003)
- Modroiu, E.R., Schieferdecker, I.: Defect Rate Profile in Large Software-Systems. In Tyugu E., Yamaguchi, T. (eds.) Proc of the 7th Joint Conference on Knowledge-Based Software Engineering. IOS Press, Amsterdam (2006)

Acronyms

- ACAP: Analyst Capability
- COCOMO II: COnstructive COst MOdel II
- COQUALMO: COnstructive QUALity MOdel
- DC: Dynamic COQUALMO
- DI: number of defects introduced
- DIR_{nom}: nominal defect introduction rate
- DR: number of defect removed
- DRF: defect removal fraction
- KSLOC: thousand source lines of code
- ODC: orthogonal defect classification
- QAF: quality adjustment factor
- $Size^B$: software size raised to a scale factor

AEROSPACE
Assuring Space Mission Success

Dynamic COQUALMO: Defect Profiling over Development Cycles

Dan Houston, Douglas Buettner, and Myron Hecht
The Aerospace Corporation

Computers and Software Division
May 16, 2009
Daniel.X.Houston@aero.org
310-336-0732