October 14, 2005

Attention:

Delma Moore
Contracting Officer, Code 210.3
NASA Software IV&V Facility
100 University Drive
Fairmont, WV 26554-8818

Wesley Sweetser
NASA Software IV&V Facility
100 University Drive
Fairmont, WV 26554-8818

Subject: Return on Investment of IV&V Phase III Study Final Report

Reference: Contract Number: GS-35F-4815G
Delivery Order: S-43619-Y, GSFC Code Y and NASA Research Center IV&V
CM Number: GSFCY & NRC IVV-05-137

Dear Ms. Moore and Mr. Sweetser:

Titan Corporation is pleased to provide the Return on Investment of IV&V Phase III Study Final Report, DID 06, approved for delivery under Task Order Number 31 Modification 3, Return on Investment for IV&V, for Contract Number GS-35F-4815G, BPA Order Number S-43619-Y, and Delivery Order 01. Enclosed is an electronic version of the Final Report for your review. This email and its attachment constitute the electronic delivery of Product GSFCY & NRC IVV-05-137.

Should you have any questions, please contact the undersigned.

Approved,

James Dabney
Principal Investigator
IV&V of GSFC Code Y and NASA Research Center Software
Titan Corporation
Phone: (281) 480-4101
Fax: (281) 480-6328

Enclosures: Return on Investment of IV&V Phase III Study Final Report

Distribution:

D. Moore/ NASA-Fairmont
W. Sweetser/NASA-Fairmont
K. McGill/ NASA Fairmont

J. Dicks/Titan
T. Mascaro/Titan
K. Williams/Titan

Contract Number: GSA-35-F-4815G S-43619 CM Number: GSFC & NRC IVV-05-137

Prepared for:

National Aeronautics and Space Administration

NASA IV&V Facility
Fairmont, WV 26554

DID Number: 06

INDEPENDENT VERIFICATION AND VALIDATION (IV&V) OF NASA PROGRAM SOFTWARE

Return on Investment of Independent Verification and Validation Study Phase III Final Report

October 14, 2005

TITAN

100 University Drive
Fairmont, WV 26554-8818

Abstract

This report documents results of the tasks associated with Phase III of the Independent Verification and Validation Return on Investment research. These tasks were to 1) develop initiating material for research to develop a full lifecycle prototype predictive ROI model, 2) produce prototype Bayesian belief network (BBN) sub-nets to model defect introduction and defect removal efficiency for IV&V and developers for the entire software lifecycle, 3) elicit PDFs for each node in the system of BBNs and 4) Calibrate the predictive model using existing case study data. The first task resulted in the development of a refined requirements phase BBN including updated pdf data. Tasks (2) and (3) were performed concurrently, resulting in complete lifecycle BBN diagrams and a software model. In Task (4) the prototype model was calibrated and produced predicted ROI results consistent with the case studies. The report concludes that the emphasis for the next phase of the ROI work should be collection of additional case studies and improved model calibration.

Table of Contents

1 Introduction

The independent verification and validation (IV&V) return on investment (ROI) study is developing the means to compute ROI for past projects and to predict ROI for new projects using available project data. The ROI study consists of a sequence of phases. Phase I entailed a set of preliminary direct ROI case studies. In Phase IIA, the team investigated the feasibility of computing indirect ROI and identified suitable project characteristics for indirect ROI and a candidate set of components of a predictive ROI model. Phase IIB produced a full lifecycle cost escalation model, proposed the Bayesian belief network (BBN) as a means to estimate defect density, and determined the sensitivity of the ROI model to variations in escalation rates and defect location rates. This report documents the results of Phase III which had four principal tasks:

1. Develop initiating material for research to develop a full lifecycle prototype predictive ROI model Sensitivity study
2. Produce prototype Bayesian belief network (BBN) sub-nets to model defect introduction and defect removal efficiency for IV&V and developers for the entire software lifecycle.
3. Elicit PDFs for each node in the system of BBNs
4. Calibrate the predictive model using existing case study data

2 Predictive Model Overview

The direct IV&V ROI model [DBO04] provides the means to compute ROI for completed IV&V projects. The model requires as inputs developer and IV&V costs (typically in equivalent person months (EPM)), software product size (measured in source lines of code (SLOC) or function points (FP)[FPUG00]), and measures of defect detection by the developer and IV&V for each type of issue (requirements, design, code, test, integration) and each development phase (typically in cost-to-fix EPM or issue size in FP). The direct ROI model is depicted graphically in Figure 1.

The complete set of inputs for the direct ROI model does not exist until the project is complete. Thus, the direct ROI model provides one measure of value added for complete projects, but does not (except reasoning by analogy) help in determining the potential value added of candidate IV&V projects. A predictive ROI model will permit assessment of ROI of candidate projects and therefore assist managers in resource allocation. A predictive model can also serve as the basis for a model-based effectiveness metric that will permit progressive monitoring of ongoing IV&V projects.

A predictive ROI model based on the direct ROI methodology must provide the means to determine early in the project all inputs to the direct ROI model. Estimates of developer and IV&V cost should be available early in the lifecycle as these estimates are normal project management requirements. Other inputs, specifically developer and IV&V defect discovery data, can't be known early in the project and must therefore be estimated using project characteristics that are known early in the lifecycle. The predictive ROI model uses the BBN technique [FKN01], [FKN01A] to estimate the inputs to the direct ROI model [DBO04] using information available early in the project lifecycle.

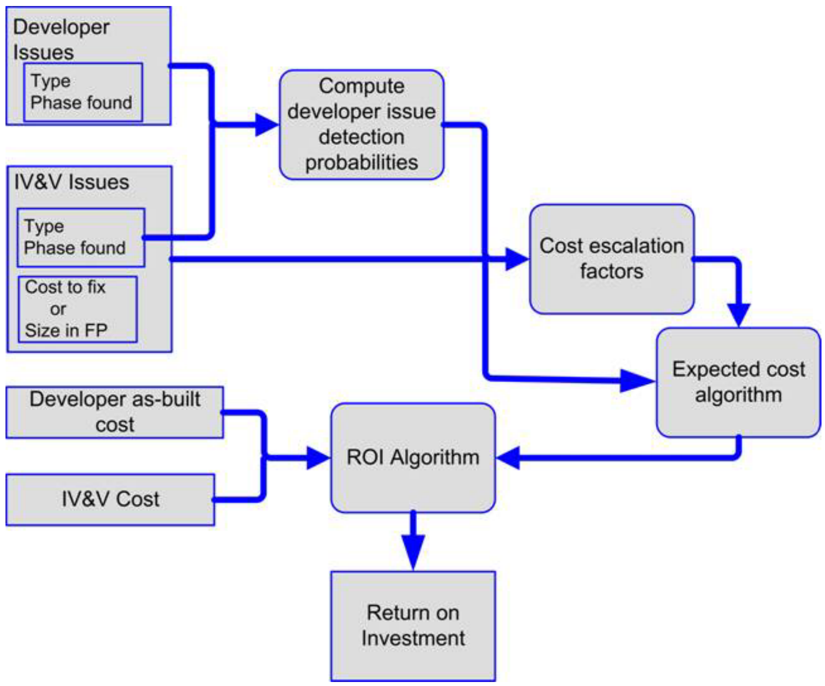


Figure 1: Direct ROI computation

2.1 Bayesian Belief Network Overview

A BBN consists of a hierarchy of nodes representing stochastic causal relationships. Figure 2 depicts a single node with two inputs. The node output is a random value with a probability density function (pdf) which depends on the values of input parameters A and B. The mapping of parameter values to pdfs could be done using historical data, if sufficient data existed. In the absence of a sufficient quantity of data, the mapping can be estimated using expert opinion. For the ROI BBN, the expert opinion approach was selected. For the ROI BBN, all nodes produce random variables in the range of 1 to 5, where 1 corresponds to the worst possible case and 5 corresponds to the best possible case.

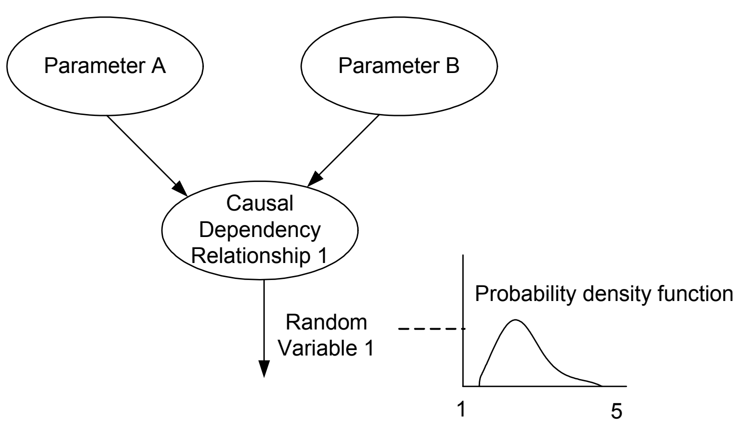


Figure 2: BBN node

A complete BBN consists of a hierarchical set of nodes. Figure 3 shows a larger BBN fragment containing three nodes. Note that the inputs to node 3 are random variables and therefore node 3 must compute an expected probability density function. In practice, this is easily accomplished using the Monte Carlo method.

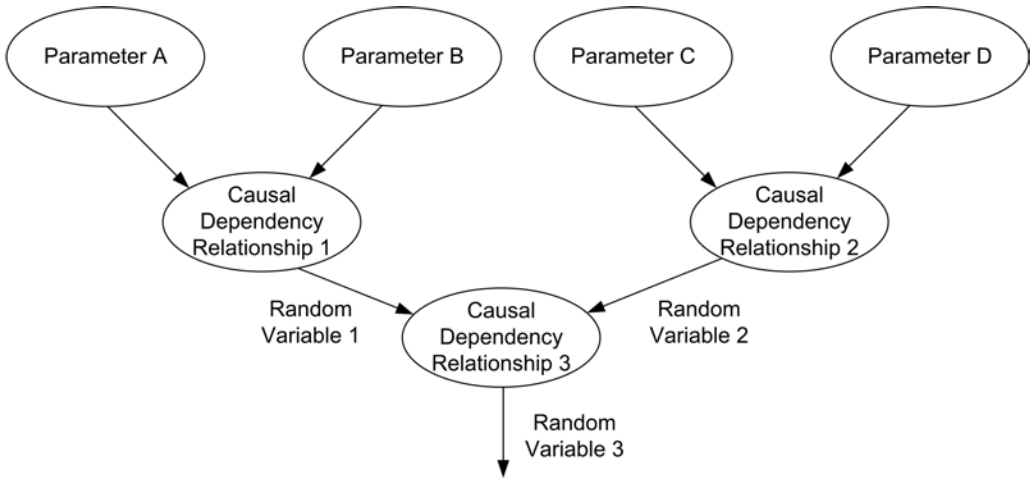


Figure 3: Hierarchical BBN nodes

2.2 Function Point Ratios

For each development phase (requirements, design, code, test, integration), three BBN subnets were developed. The first subnet estimates product quality (with respect to defect density) on a scale of 1 to 5. The second subnet estimates developer defect detection efficiency on a scale of 1 to 5. The third subnet estimates IV&V defect detection efficiency on a scale of 1 to 5. In order to produce the inputs required by the direct ROI model, the BBN output must be converted into measures of defect size. Function points are a convenient measure early in the lifecycle because they can be estimated from system requirements and are independent of programming language. Therefore the subnet values (product quality, developer defect discovery efficiency , IV&V defect discovery efficiency) are used to estimate function point ratios (FPRs), which, scaled by the estimated product function points, provide suitable direct ROI model inputs. A high level flow diagram for the process (showing requirements phase defect function points only) is shown in Figure 4.

The overall process of developing the BBN model for one phase (requirements, design, code, test, integration) of the baseline IV&V process consists of the following steps:

1. Develop pictorially (based on elicitation from experts) the BBNs for defect introduction, defect detection by the developer, and defect detection by IV&V.
2. Elicit from the experts probability density functions for each causal dependency.
3. Implement the BBN in software using the Monte Carlo technique.
4. Calibrate the BBN output to case study data to predict discovered defect function points in-phase for the developer and IV&V

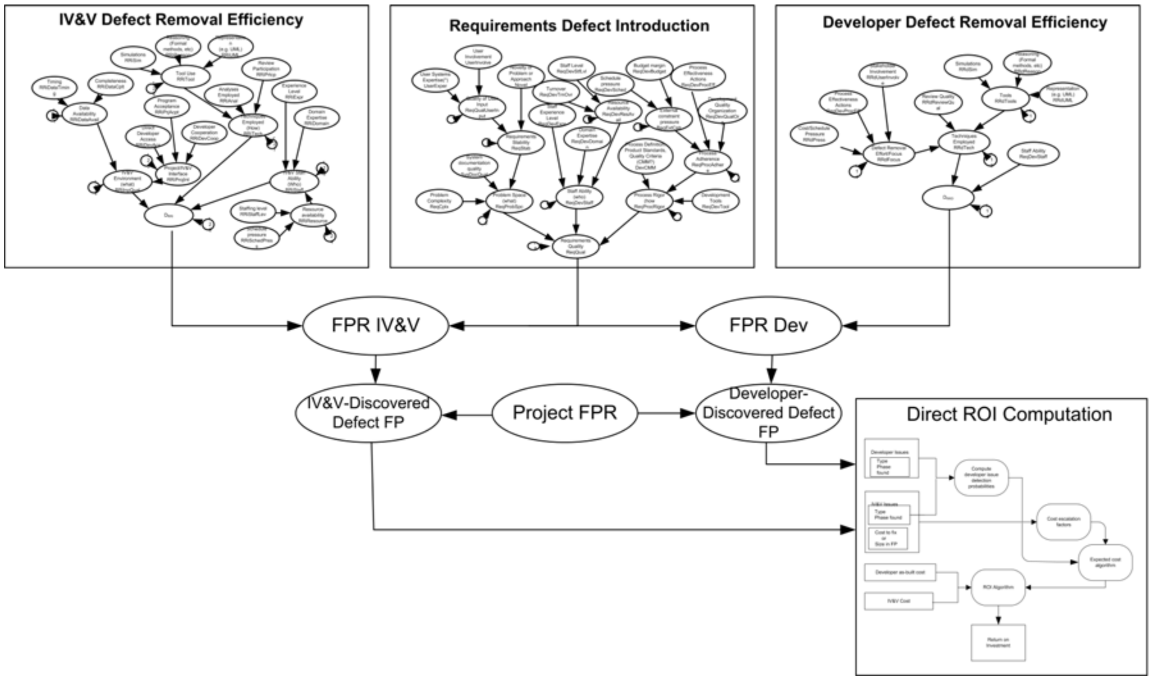


Figure 4: Predictive model defect function point computation

2.3 Defect Leakage Model

The direct ROI model requires as inputs discovered-defect function points for the developer and IV&V for each issue type and phase found. The BBNs were developed to compute in-phase discovered defects only. Although it would be possible to develop a BBN for each issue type for each development phase, that would require an additional fifteen subnets many of which would lack calibration data. Therefore, a leakage model was devised to estimate discovered defect function points out of phase (for example, requirements defect function points discovered in design, code, test, integration phases). Based on an extensive literature search, the Rayleigh leakage model [GJWE01], [KSH91] was selected. Calibration of the leakage model will be discussed later.

3 Complete Prototype BBN

The complete BBN framework consists of subnets for defect introduction and defect detection by the developer and IV&V for each development phase. As noted above, BBNs are used only for in-phase defect detection prediction. Defect discovery in subsequent phases is estimated using a calibrated Rayleigh leakage model. Table 1 lists the source of direct model inputs for each phase and issue type. Here, BBN indicates the source is the BBN and LM indicates the source is the calibrated Rayleigh leakage model applied to the BBN FPRs.

Table 1: Function point ratio source for ROI computation

Issue type	Phase issue found					
	Requirements	**Design**	**Code**	**Test**	**Int**	**Ops**
Requirements	BBN	LM	LM	LM	LM	LM
Design		BBN	LM	LM	LM	LM
Code			BBN	LM	LM	LM
Test				BBN	LM	LM
Integration					BBN	LM

3.1 BBN Subnets

Appendix A shows the complete set of BBNs. An example defect introduction subnet (requirements phase) is shown in Figure 5. The output of this subnet is requirements quality in the range of 1 to 5 where 1 is the worst possible quality and 5 is the best possible quality. An example defect detection BBN (IV&V, requirements phase) is shown in Figure 6. The output of the defect detection subnet is the ratio (FPR) of function points of defects discovered to total function points. The product of FPR and total function points is a suitable set of inputs for the direct ROI model.

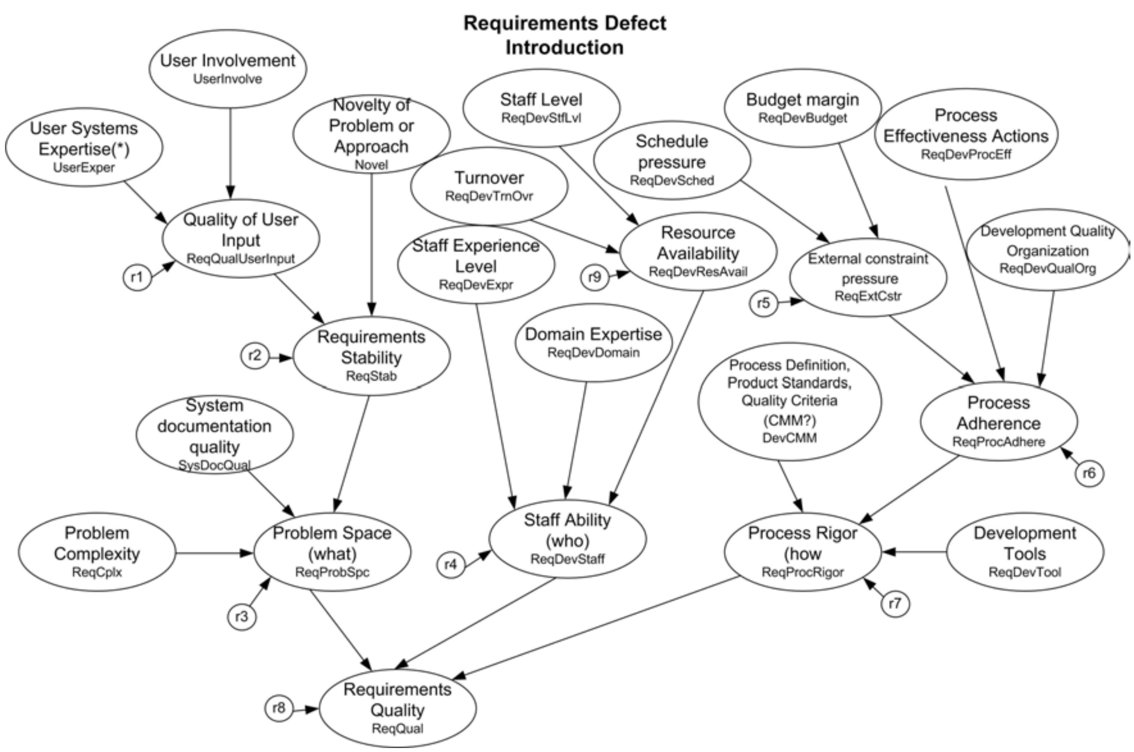


Figure 5: Requirements quality BBN sub-net

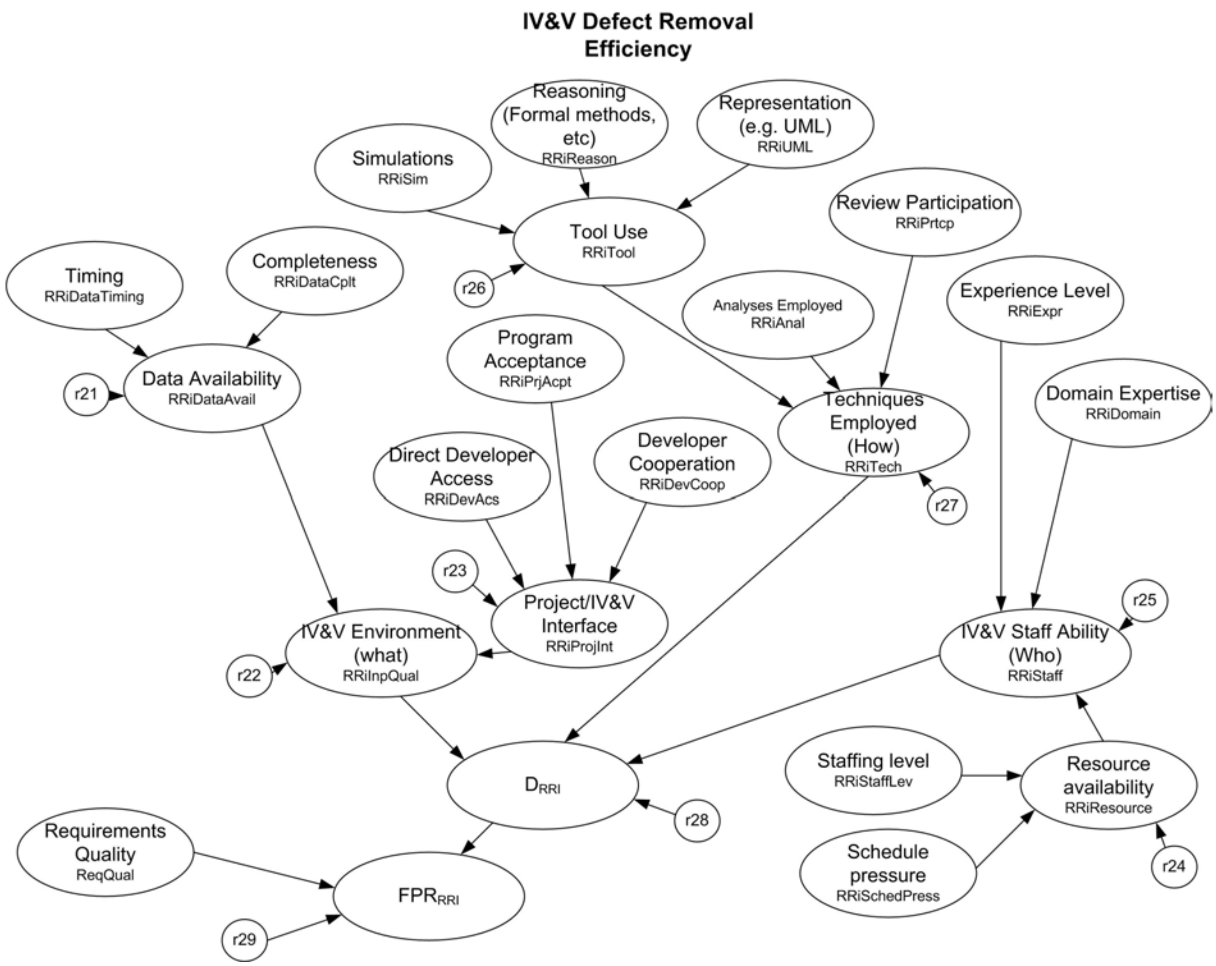


Figure 6: IV&V requirements phase defect detection sub-net

3.2 Monte Carlo Implementation

The complete set of BBN sub-nets was implemented in MATLAB using a Monte Carlo technique. For the prototype software, each phase is implemented as a stand-alone MATLAB program. Inputs for subsequent phases from BBNs earlier in the lifecycle are manually entered in the prototype model. The number of Monte Carlo iterations is an input parameter; using 100,000 iterations was found to produce stable results. The structure of each BBN subnet model is as follows:

- Load BBN input data (elicited project characteristics, see section 5.1)
- Monte Carlo iteration loop
  - For each node in succession
    - Interpolate in the node pdf tables to generate a pdf corresponding to the node inputs. Details of the pdf representation are presented in the next section.

        - Using the MATLAB random number function and the pdf from the previous step, generate a node output
    - Store the iteration results
- Compute expected value and standard deviation for each node output
- Generate plots and print results

4 Node Probability Density Functions

4.1 PDF Representation

Each node in the ROI BBN produces a random variable between 1 and 5 with a pdf that depends on the node inputs. In order to avoid excessive complexity, each node has either two or three inputs. More complex relationships are represented by cascading the nodes. The node pdf functions are based on a set of elicited pdfs for each node. The pdfs are elicited from experts using a pdf editor graphical user interface (GUI). The pdf editor window for a typical node with two inputs is shown in Figure 7.

The pdfs are elicited for boundary cases and internal cases. Each pdf is approximated using a trapezoidal distribution by dragging the circles that correspond to the four points in the pdf. Thus, a pdf corresponding to a particular node input vector is approximated by the location of the four points that define the trapezoid. In order to implement the BBN in software, it is necessary to map the node inputs to the locations of the four points that characterize the node pdf. The location of each pdf point for a node with two inputs is a surface as shown in Figure 8, and the location of each pdf point for a three-input node is a hypersurface.

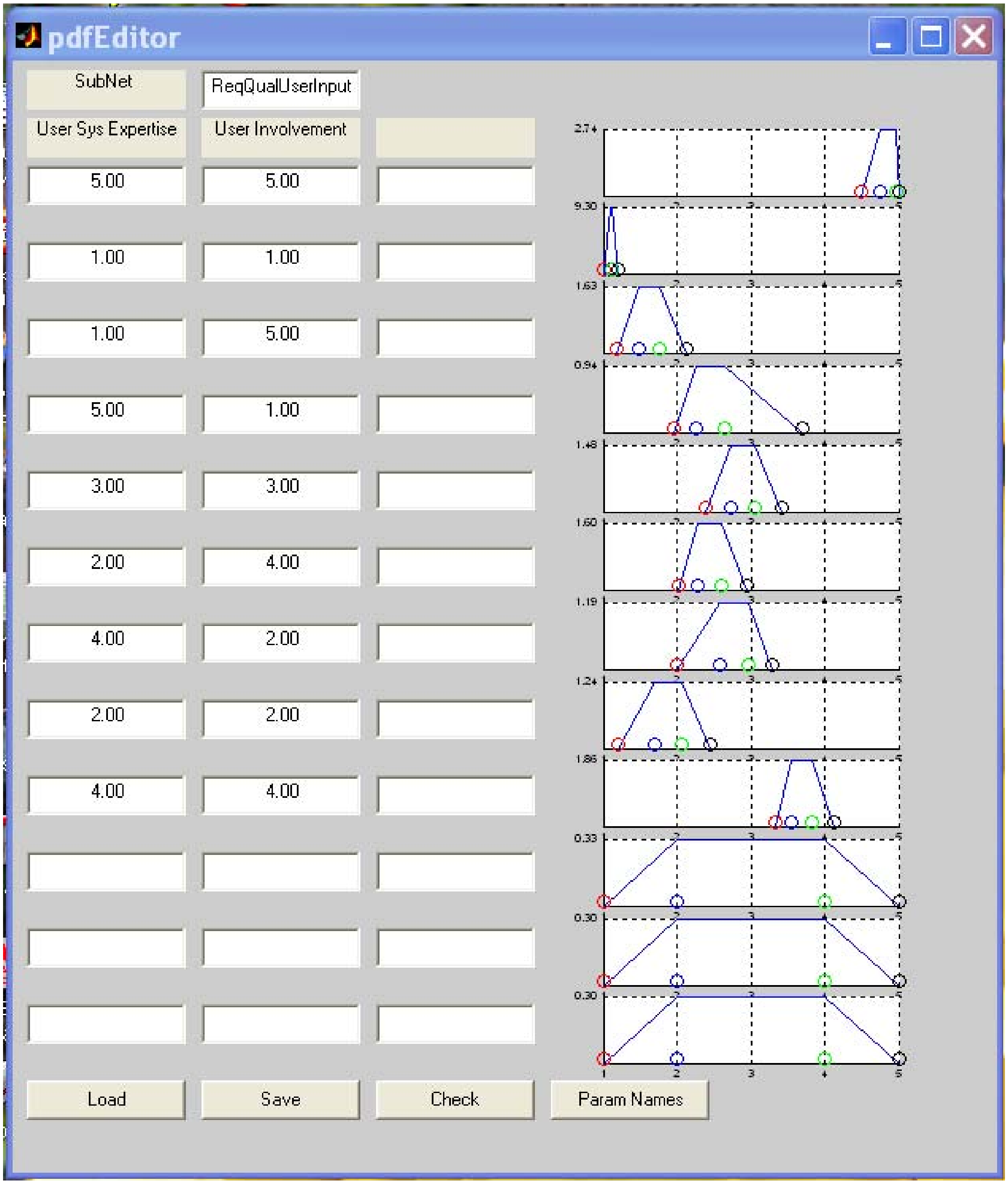


Figure 7: pdf editor GUI

A number of techniques for extending the elicited pdfs to node functions were evaluated. Most of the techniques used curve fitting to approximate the functions for each node point. None of the curve fitting techniques produced consistently satisfactory results. Therefore, the node pdfs are represented as lookup tables that are produced by interpolating and extrapolating in the elicited data to generate a grid of points that are compatible with the built-in MATLAB two- and three- dimensional interpolating functions. The surface shown n Figure 8 was produced from the interpolating table for the second point of the node pdf function defined in Figure 7.

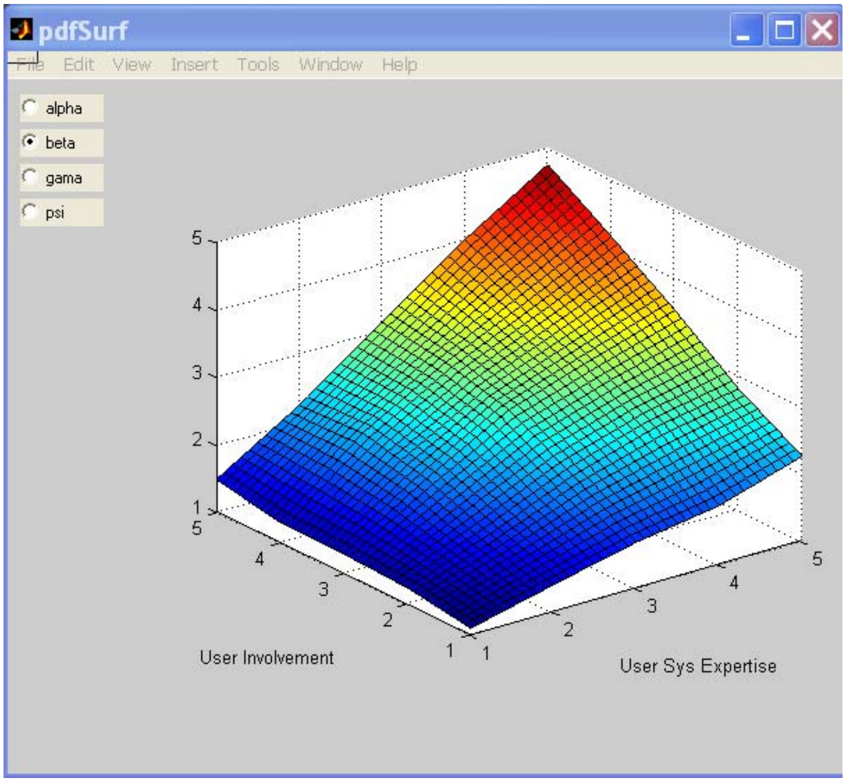


Figure 8: Interpolating surface

4.2 PDF Elicitation

Once the complete set of BBN subnets was defined, the pdf editor (Figure 7) was used to capture elicited pdfs for each node. The pdfs were validated using the plotting feature illustrated in Figure 8.

5 Model Calibration

After eliciting pdf functions, it was necessary to calibrate the predictive model to the case study data. The calibration was performed in four steps. First, BBN input data was collected from project managers for each of the four direct ROI case studies. FPRs were calculated for in-phase and leakage issues using the data collected for the direct ROI case studies. Next, FPR functions were developed for in-phase issues for each defect type (requirements, design, code, test, integration) for developer and IV&V issues. Last, the leakage model was calibrated to predict developer and IV&V defect detection in subsequent life cycle phases. Using the calibrated model, ROI was predicted for each of the four case studies and compared to case study results.

5.1 BBN Input Data Collection

The prototype predictive model was calibrated using the four direct ROI case studies previously reported [DBO03]. Inputs for each BBN sub-net for each of the four case studies were collected from IV&V project managers familiar with the case study projects. The inputs were collected using spreadsheets with embedded instructions. The spreadsheets included internal nodes to aid in validating the BBN topology and to ensure consistent data. Inconsistencies between internal nodes were discussed with the project managers and adjustments made to achieve consistent input. The spreadsheets were configured to automatically generate the MATLAB code. The score for each node is an estimated value in the range 1 - 10, a lower tolerance, and an upper tolerance. The range 1 – 10 was chosen because preliminary experiments indicated that eliciting inputs in the range 1 – 5 provided insufficient discrimination among projects. The spreadsheet generates automatically the MATLAB code that interprets the inputs as triangular probability density functions as illustrated in Figure 9.

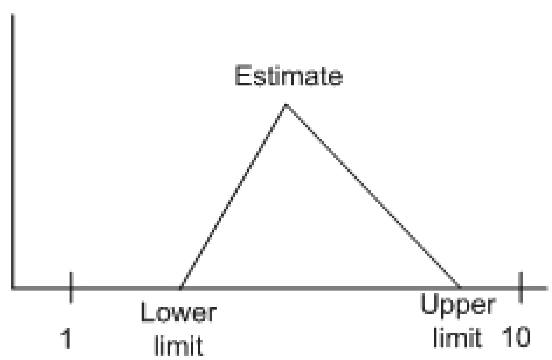


Figure 9: Node input pdf

The MATLAB code then rescales the pdfs to the 1 – 5 range used in the BBN. Appendix B contains the input descriptions contained in the spreadsheet files for each issue type.

5.2 FPR Calibration

The FPR calibration mapped the three BBN outputs (quality Q_{ϕ}, developer defect discovery efficiency $D_{\phi\phi d}$ and IV&V defect removal efficiency $D_{\phi\phi i}$, where ϕ represents the development phase and issue type). The mapping functions consist of lookup tables for each phase for developer and IV&V that map quality and efficiency to FPR. The main steps in the calibration for each issue type were as follows:

- Run the BBN model for each case study using the elicited BBN inputs
- Convert the case study defect data (actual data) to FPR
- Plot the actual FPR as a function of quality and defect discovery efficiency to identify candidate approximating functions
- Fit approximating functions to the BBN output and case study FPR
- Generate FPR lookup tables suitable for MATLAB cubic spline interpolation and implementation in the MATLAB BBN models

- Re-run the BBN using the calibrated FPR functions and compare computed FPR with actual FPR.

5.2.1 Developer-Discovered FPR Calibration

For developer-discovered issues, plotting actual FPR vs requirements quality and developer defect removal efficiency, it was observed that FPR varies directly with the distance from the origin in the (Q_R, D_{rrd}) plane. Therefore, an approximating function of the form

$$FPR_{RRd} = c_{RRd}\sqrt{Q_R^2 + D_{RRd}^2}$$

was used. The calibrated curve is shown in Figure 10. Here the circles are the case study data points and the solid trace is the approximating function. The corresponding interpolating surface is shown in Figure 11.

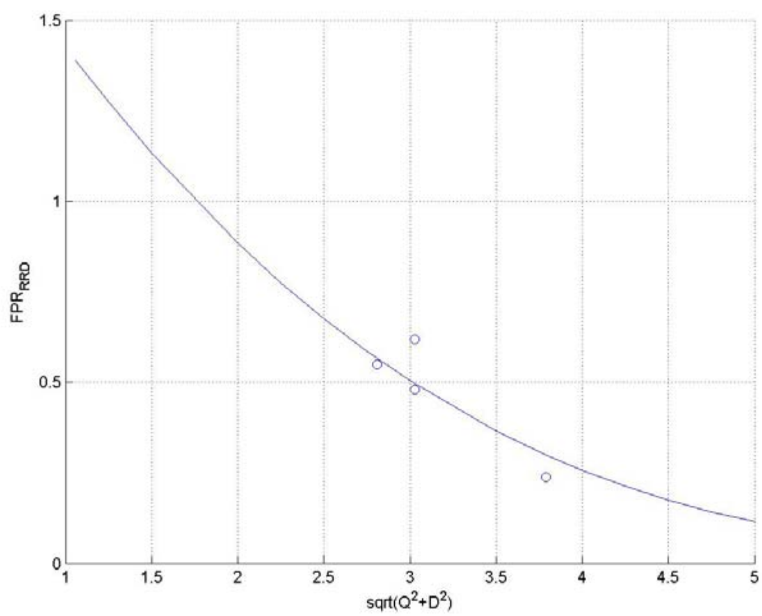


Figure 10: Requirements phase FPR_{DDd} calibration

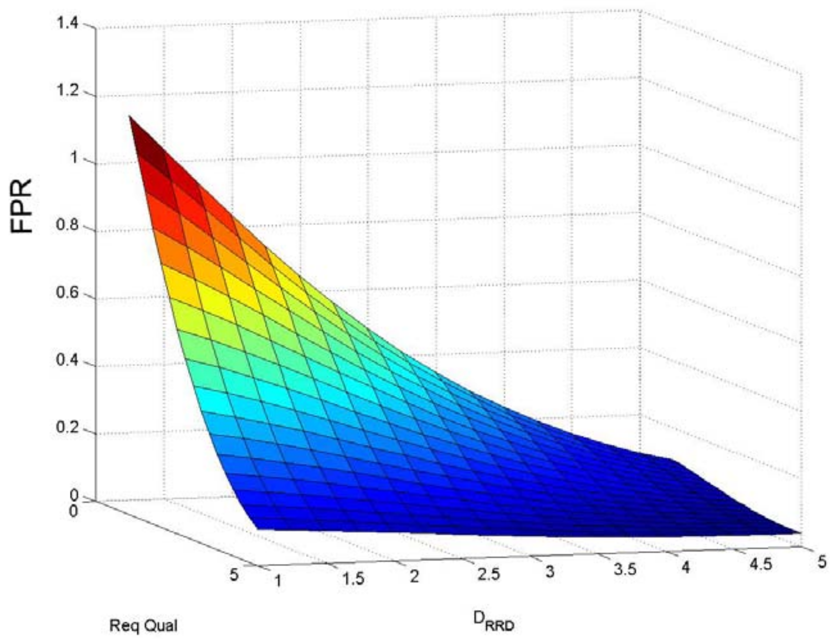


Figure 11: FPR_{RRd} interpolating surface

5.2.2 IV&V-Discovered FPR Calibration

For IV&V-discovered issues, a more complex relationship was postulated and observed to fit well to the four case studies. The defect detection opportunity was postulated to be a function of Q_ϕ such that there is an optimal point that depends on issue type. For Q_ϕ values lower than the optimal point, the defect discovery opportunity is reduced because defect discovery is inherently more difficult. For Q_ϕ values greater than the optimal point, defect discovery opportunity is reduced because there should be fewer defects to discover. It was also observed that the effectiveness of IV&V defect discovery exhibits an inverse exponential relationship with IV&V efficiency score. The best fit for IV&V effectiveness was a function of the form

$$Eff_{IV\&V} = a\, e^{-bD_{\phi\phi i}} \cos(c\, D_{\phi\phi i})$$

where a, b, c are coefficients determined using a nonlinear least squares technique. Due to the difficulty in fitting functions to the opportunity data, interpolating tables were produced graphically and used to generate interpolating surfaces for FPR. An example IV&V FPR interpolating surface is shown in Figure 12.

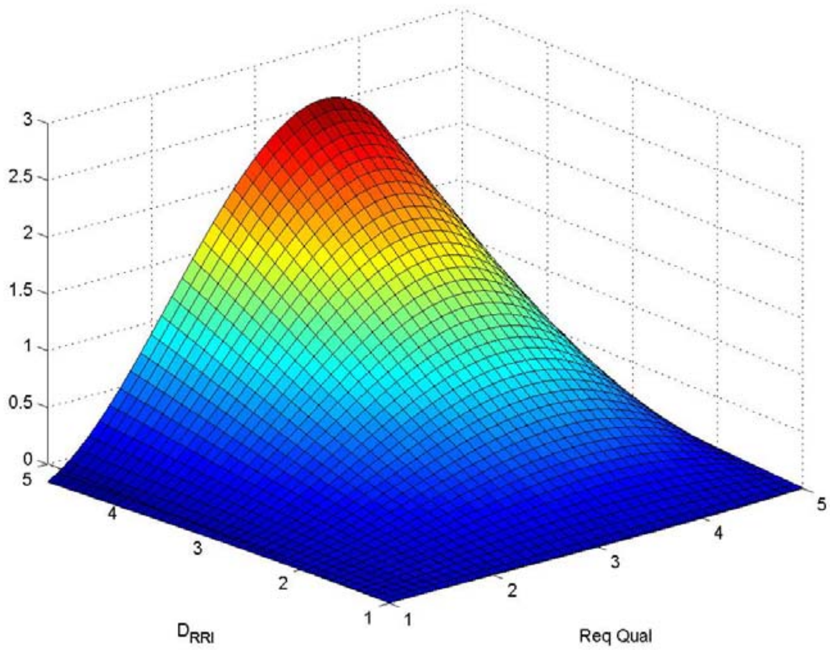


Figure 12: FPR_{RRi} interpolating surface

5.2.3 FPR Calibration Results

The results of the in-phase issue detection FPR calibration are shown in Tables 2 - 6. For requirements defects, the case study data was consistent with the Q_ϕ and D_{rrx} results for all projects. Consequently, the predicted FPRs agree within one standard deviation with the actual FPRs. For subsequent life cycle phases, there are outliers that resulted from project anomalies. For example, project B produced no design documentation although many characteristics of the developer's process were judged by the IV&V project manager to be relatively good. Project C was terminated before any code, test, or integration issues were reported, suppressing IV&V defect FPR even though a relatively good IV&V process was underway when the development project was terminated. For all of the case study projects, it is evident that in-phase issue reporting for the later lifecycle phases tended to be lower than earlier lifecycle phases. Of course, due to the relatively small cost escalation factors for the later lifecycle phases, this factor will have a relatively small impact on ROI.

Table 2: Requirements phase FPR calibration

Case	Developer FPR			IV&V FPR		
	Actual	**Predicted**	**Std Dev**	**Actual**	**Predicted**	**Std Dev**
A	0.551	0.709	0.166	0.096	0.135	0.098
B	0.483	0.629	0.176	0.079	0.093	0.070
C	0.239	0.371	0.135	0.812	0.603	0.286
D	0.623	0.628	0.161	0.137	0.289	0.169

Table 3: Design phase FPR calibration

Case	Developer FPR			IV&V FPR		
	Actual	**Predicted**	**Std Dev**	**Actual**	**Predicted**	**Std Dev**
A	0.429	0.541	0.114	0.429	0.457	0.593
B	0.000	0.504	0.135	0	0.210	0.290
C	0.037	0.321	0.104	2.207	2.105	1.343
D	0.451	0.433	0.114	0.712	1.510	1.305

Table 4: Code phase FPR calibration

Case	Developer FPR			IV&V FPR		
	Actual	**Predicted**	**Std Dev**	**Actual**	**Predicted**	**Std Dev**
A	3.700	3.328	2.027	1.821	0.821	0.388
B	1.512	2.090	1.502	0.485	1.026	0.490
C	0.037	1.252	0.869	0	1.062	0.717
D	0.107	2.814	1.535	0.040	1.163	0.555

Table 5: Test phase FPR calibration

Case	Developer FPR			IV&V FPR		
	Actual	**Predicted**	**Std Dev**	**Actual**	**Predicted**	**Std Dev**
A	3.323	1.069	0.390	1.640	0.679	0.236
B	1.292	1.547	0.415	0	0.347	0.180
C	0.075	0.669	0.305	0	0.589	0.245
D	0.002	0.622	0.290	0.086	0.597	0.230

Table 6: Integration phase FPR calibration

Case	Developer FPR			IV&V FPR		
	Actual	**Predicted**	**Std Dev**	**Actual**	**Predicted**	**Std Dev**
A	0	0.019	0.005	0	0.0004	0.0002
B	0	0.018	0.005	0	0.0002	0.0001
C	0.037	0.007	0.004	0	0.0003	0.0002
D	0.002	0.011	0.005	0.003	0.0005	0.0002

5.2.4 Leakage Model Calibration

The leakage model was calibrated using only two of the four case studies. The other two case studies reported no defect leakage to subsequent life cycle phases. The lack of leakage data for the two projects resulted from project anomalies rather than the lack of defect leakage. One project did not track post-phase issues and for the other project, the case study was based on a database snapshot that didn't include the leakage data for IV&V and for which case study developer leakage data was estimated by IV&V project managers due to the lack of actual developer data.

Using averages of the available leakage data, the Rayleigh model was calibrated for IV&V and developer-discovered defects by fitting the cumulative issues to the Rayleigh function. The results of developer calibration are shown in Figure 13 and the results of IV&V calibration are shown in Figure 14. In both cases, the cumulative FPR is normalized to the in-phase data so that the computed issues in each subsequent phase is the product of the leakage factor and in-phase FPR. In the figures, the small circles represent the case study data and the solid lines represent the leakage model.

The Rayleigh leakage model has been shown to be effective for large numbers of projects. However, it was observed that leakage exhibits rather large variances. With only two case studies upon which to calibrate leakage, it was not possible to compute the standard deviations of the estimates. Therefore, for the prototype model, it was assumed that the leakage standard deviation is proportional to the standard deviation for in-phase issues of each issue type. That is, the standard deviation of predicted FPR for a particular issue type for subsequent phases is assumed to be the product of the in-phase standard deviation and the ratio of predicted leakage FPR and in-phase FPR.

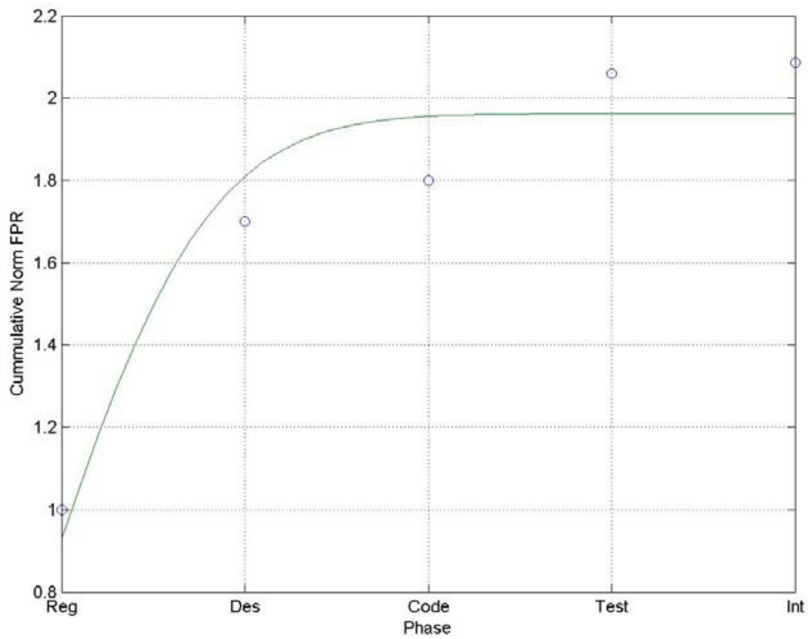


Figure 13: Leakage model calibration for developer-discovered defects

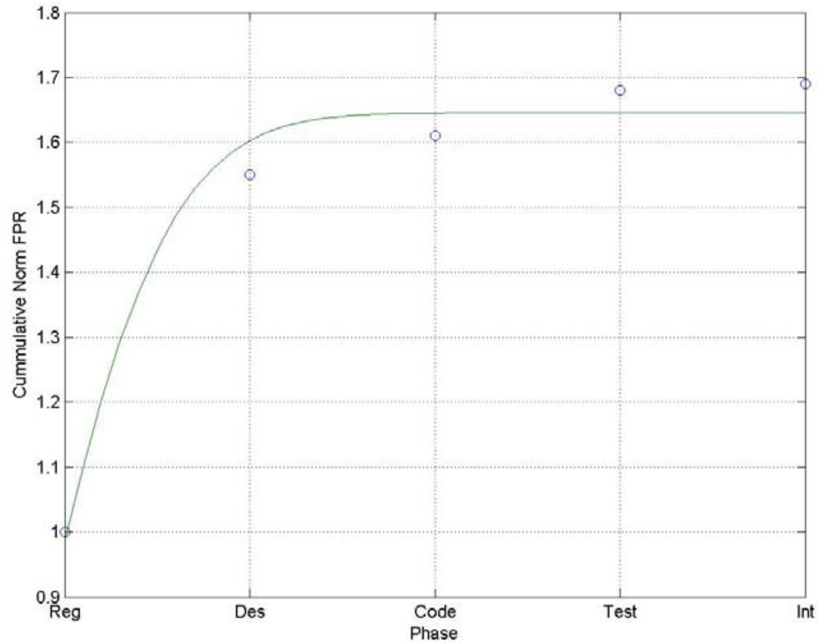


Figure 14: Leakage model calibration for IV&V-discovered defects

5.3 ROI Computation

Using the calibrated prototype predictive model, ROI was computed for each of the four case studies. ROI computations were made using the subnet results illustrated in Figure 4. The direct ROI model [DBO04], [DBO03], implemented in MATLAB, used the FPR results of Tables 2 – 6 and the leakage model of Figure 13 – 14 to estimate developer defect discovery probabilities for each Monte Carlo iteration and then to compute expected without-IV&V BRAK. Next, the COCOMO [BAB00] coefficient was calibrated to the with-IV&V results and then used to compute without-IV&V results. Using the with- and BBN-predicted without-IV&V development costs and the actual IV&V cost, ROI was computed. The results of the ROI computation are listed in Table 7.

For two cases (B & C) , predicted ROI is somewhat larger than actual ROI. In both cases, this was due to the fact that BBN predicted in-phase issues when the case study data contained no in-phase issues. To more accurately portray the actual case study situation in those cases, the elicited IV&V data for phases in which there were no IV&V issues due to lack of data availability, all IV&V inputs for those phases were set to 1.0 with zero tolerance and the BBN was then re-run. Using the corrected IV&V input data, predicted ROI for both cases is closer to actual ROI.

Table 7: Predicted ROI results

Project	**Actual Direct ROI**	**Predicted Direct ROI**	**ROI Std Deviation**
A	1.590	1.261	0.170
B	1.207	4.040 * 3.310	0.625 * 0.550
C	5.534	10.78 * 7.970	1.590 * 1.330
D	10.085	10.68	1.77

Among the four case studies, Case D had the most complete data for both IV&V and the developer. There were reported issues for both IV&V and the developer for all lifecycle phases and the project was completed successfully. Therefore, it is not a coincidence that the best agreement between actual and predicted direct ROI was exhibited by Case D.

The actual ROI computation for Case B was the least certain. Developer issue distribution among phases was not available when the case study was performed, so a conservative leakage model (developer issues distributed equally among the phases) was assumed. Due to the lack of design documentation, the developer found no design defects and IV&V found design defects only in the code phase via source code analysis. Furthermore, IV&V was not complete when the case study was performed, so no IV&V issues found in the test or integration phase were included in the original case study. It is reasonable to expect that full-lifecycle IV&V and elimination of the conservative (from the IV&V ROI perspective) assumptions would increase Case B ROI significantly.

6 Summary

The prototype predictive ROI model was developed as planned and node probability density functions (pdfs) were developed using the pdf editor. The predictive model for each BBN subnet was implemented in MATLAB. The model was calibrated using case study data to compute function point ratios (FPRs) for developer and IV&V-discovered in-phase issues. A Rayleigh defect leakage model was calibrated to the developer and IV&V case study data to predict out-of-phase FPRs. ROI was computed using a

* IV&V inputs for phases for which there was no IV&V activity set to 1.0 (minimum value)

MATLAB Monte Carlo implementation of the direct ROI algorithm and BBN-predicted ROI was computed for each of the four case studies.

The in-phase FPR computations for requirements and design phases for both developer and IV&V issues are in excellent agreement with actual data except for two easily understood anomalies in the case study data. Agreement between predicted and actual values decreases progressively as the lifecycle proceeds to code, test, and integration phases. This decline in precision of the predictive model for later lifecycle phases is attributed to decreasing availability of case study data for the later lifecycle phases. The decline in precision is ameliorated by the decreasing importance, in the direct ROI sense, of the later lifecycle phases.

The fidelity of post-phase (leakage) defect FPR is lower than in-phase fidelity due to the apparently higher variability of leakage behavior and limited amount of calibration data. As was shown in the Phase IIB sensitivity study [DBO04A], ROI is highly dependent on leakage rates because estimation of post-phase developer defect detection drives the probability distribution across lifecycle phases and therefore the expected value of cost-to-fix escalation. For example, the sensitivity study showed that for a hypothetical project with an IV&V ROI of 8.5, significantly reducing developer defect detection efficiency (or increasing leakage rate) can increase IV&V ROI to 25.3. Therefore, the variation in ROI exhibited by the predictive model is well within the envelope to be expected from the sensitivity study results.

7 Conclusions and Recommendations

The predictive IV&V ROI model produces credible ROI estimates for the four case studies. The initial calibration predicts potential full-lifecycle ROI more accurately than truncated-lifecycle ROI. Therefore, it appears that the predictive model is particularly well-suited to prediction of achievable ROI for a specified set of project circumstances. Thus, the prototype predictive model appears to be particularly well-suited to use in a model-based effectiveness measurement framework.

The prototype predictive model also suggests that although per-issue ROI is higher for early lifecycle activities, overall ROI is better for full-lifecycle IV&V. The prototype model provides the means to further explore this phenomenon via additional cases using hypothetical IV&V projects.

Proposed future work includes additional case studies and development of a production ROI model. The results of the prototype model calibration suggest that initial emphasis should be placed on expanding the calibration database via additional case studies. The additional case studies will provide more insight into IV&V ROI, will improve the model calibration database, and will serve as the basis for automating ROI data collection for in-progress projects. Additional experience working with the prototype model in the process of doing the additional case studies will facilitate improved production model requirements.

8 References

[DBO04] J. B. Dabney, G. Barber, and D. Ohi, "Estimating direct return on investment of independent verification and validation," 8th IASTED International Conference on Software Engineering and Applications, Cambridge, MA, 2004.

[FKN01] N. Fenton, P. Krause, and M. Neil, "Software measurement: Uncertainty and causal modeling," 2001.

[FKN01A] N. Fenton, P. Krause, M. Neil, "A probabilistic model for software defect prediction," preprint, University of London, England, 2001.

[DBO04A] J. B. Dabney, G. Barber, and D. Ohi, *Return on Investment of Independent Verification and Validation Study Phase IIB Final Report*, Titan Inc., Fairmont, WV, 2004.

[FPUG00] *Function Point Counting Practices Manual*, Release 4.1.1, The International Function Point User's Group, 2000.

[GJWE01] J. W. E. Greene, "Purchasing Software Intensive Systems Using Quality Targets", Quality Software Management Ltd., 2001

[KSH91] S. H Kan, "Modeling And Software Development Quality", IBM Systems Journal, Vol 30, No. 3, 1991

[DBO03] J. B. Dabney, G. Barber, and D. Ohi, Computing Direct Return on Investment of Software Independent Verification and Validation, Titan Inc., Fairmont, WV, 2003.

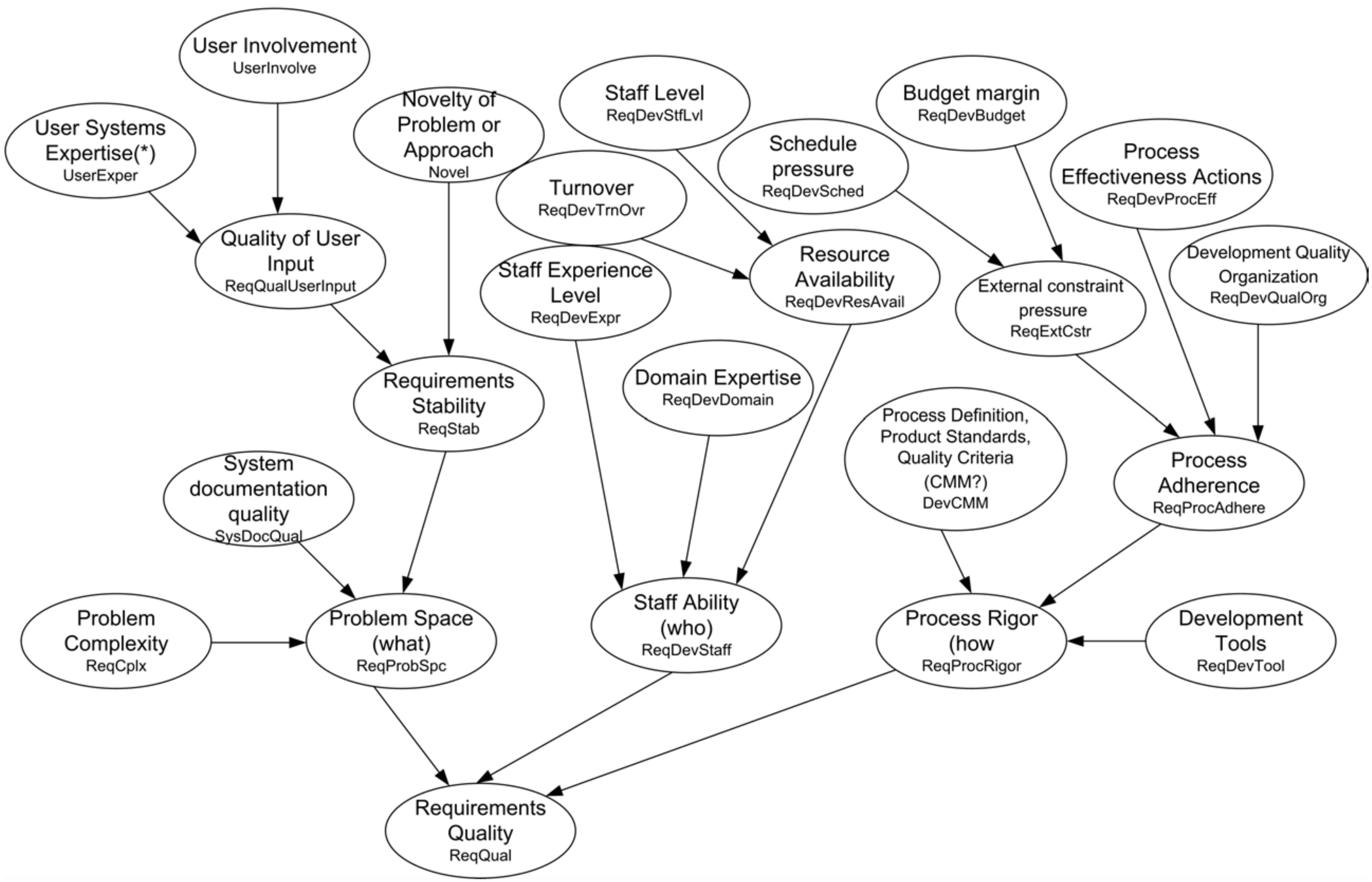
[BAB00] B. Boehm, C. Abts, A. W. Brown, S. Chulani, B. Clark, E. Horowitz, R. Madachy, D. Reifer, B. Steece, *Software Cost Estimation with COCOMO II*, Prentice Hall, Upper Saddle River, NJ, 2000.

Appendix A – BBN Diagrams

This appendix contains the complete set of BBN diagrams for each phase.

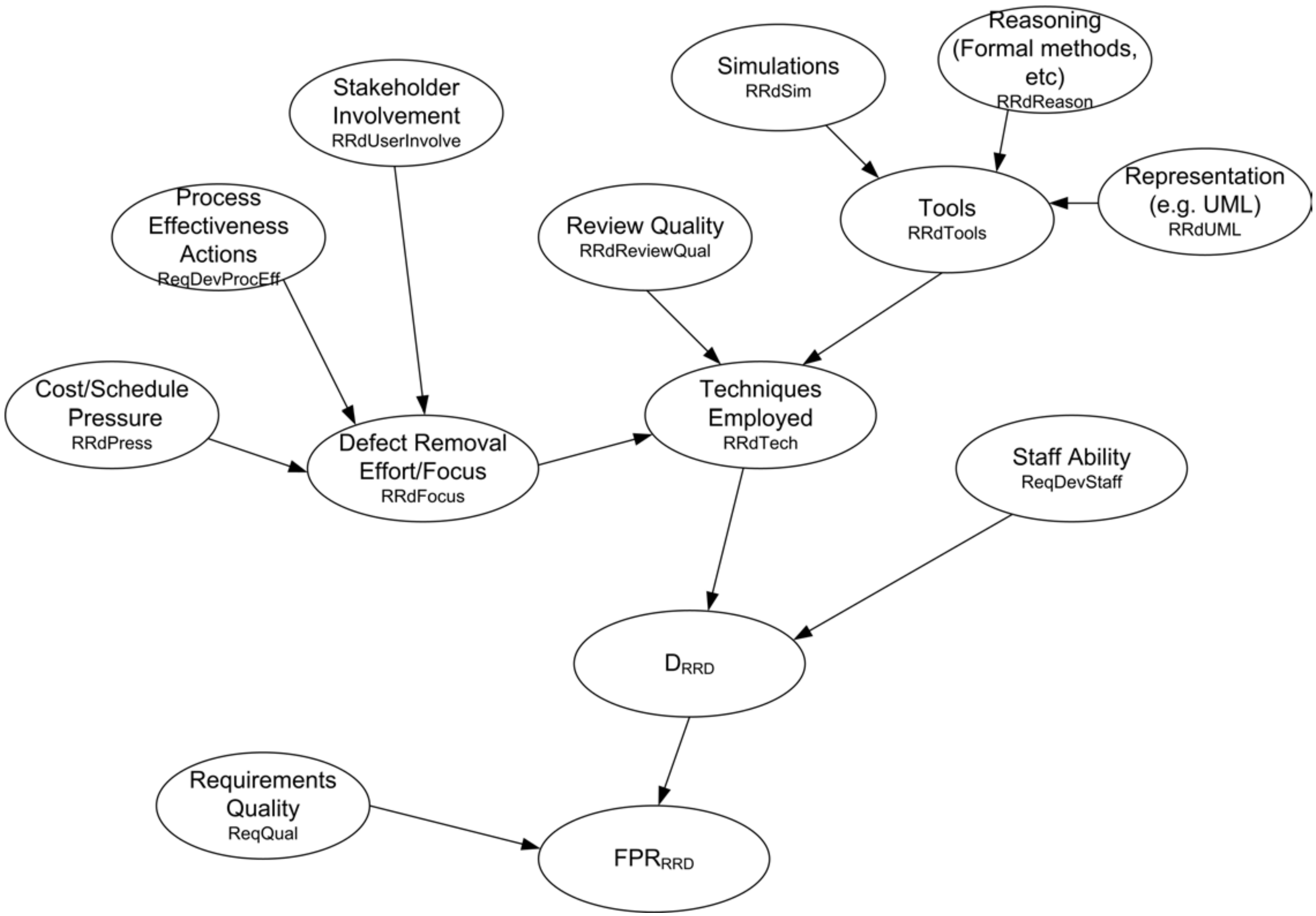
8 Oct 05

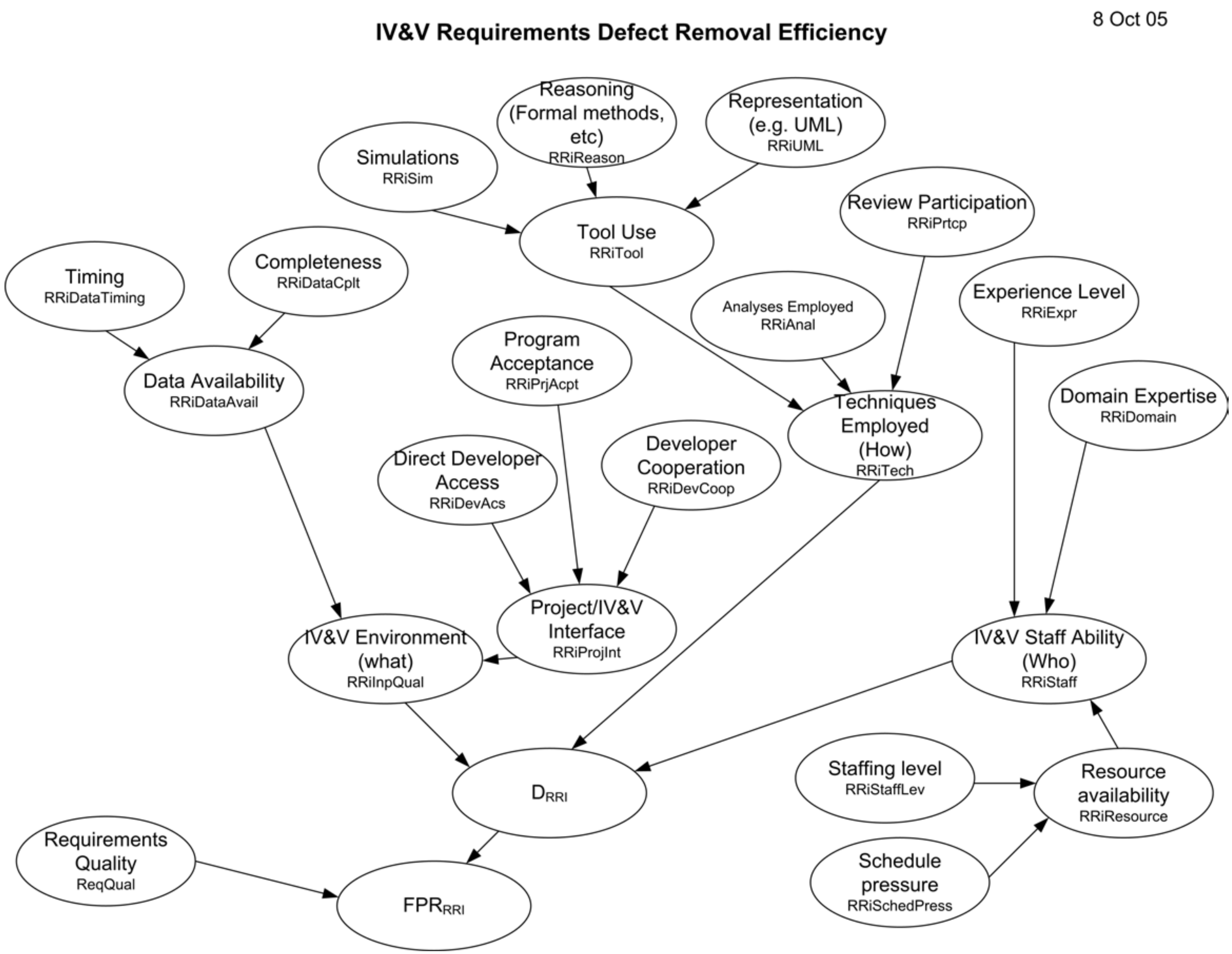
Requirements Defect Introduction

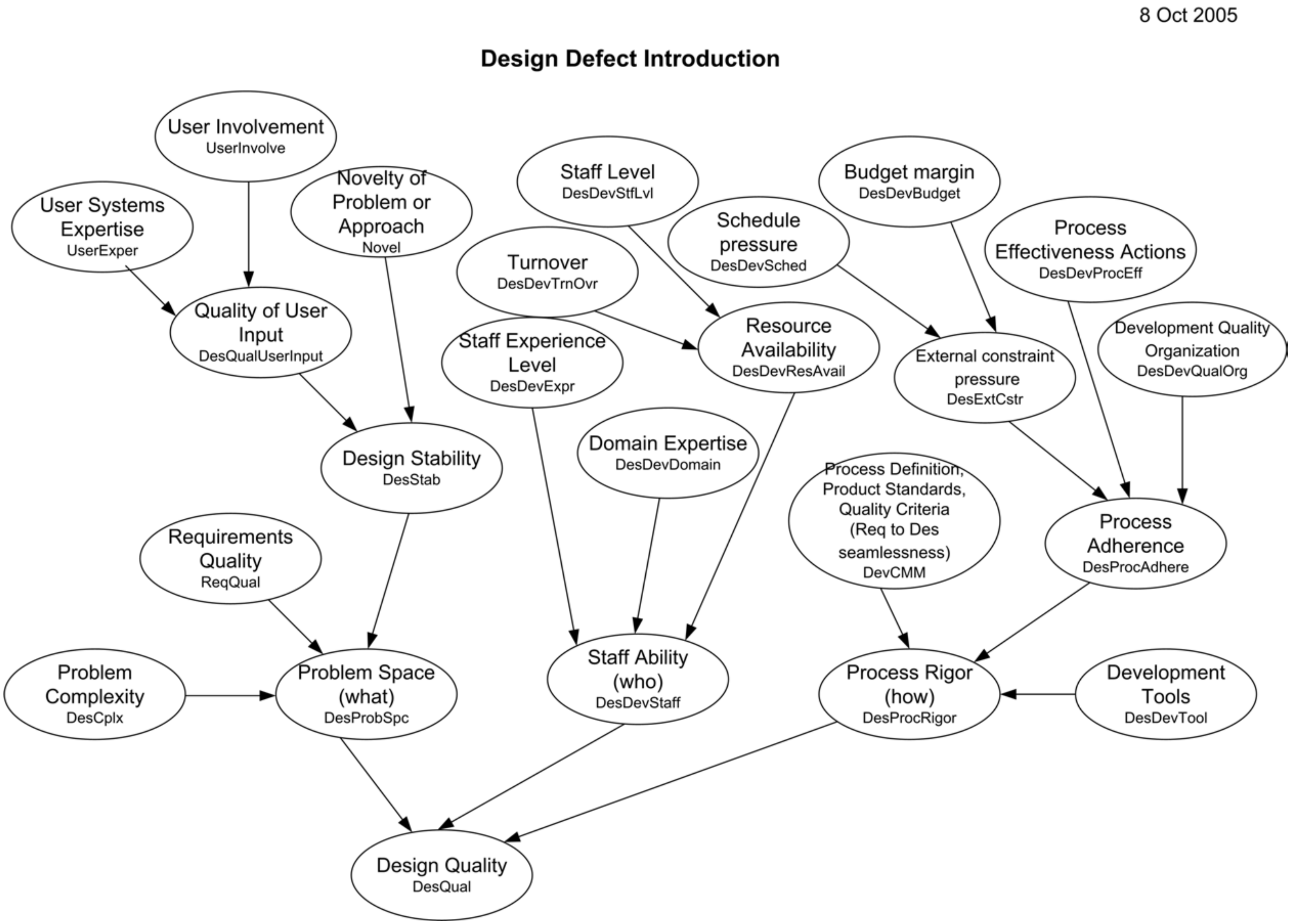


8 Oct 05

Developer Requirements Defect Removal Efficiency

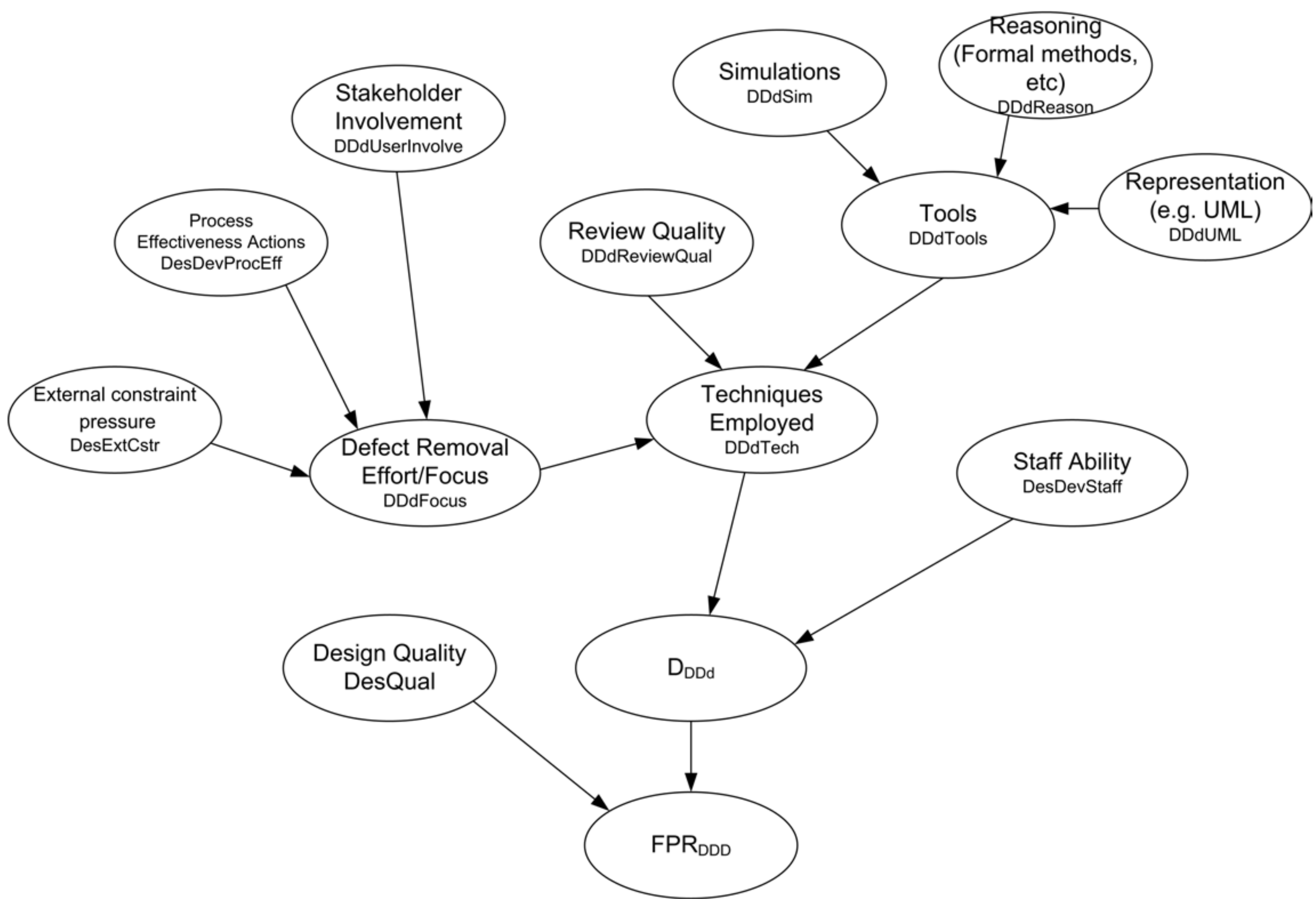


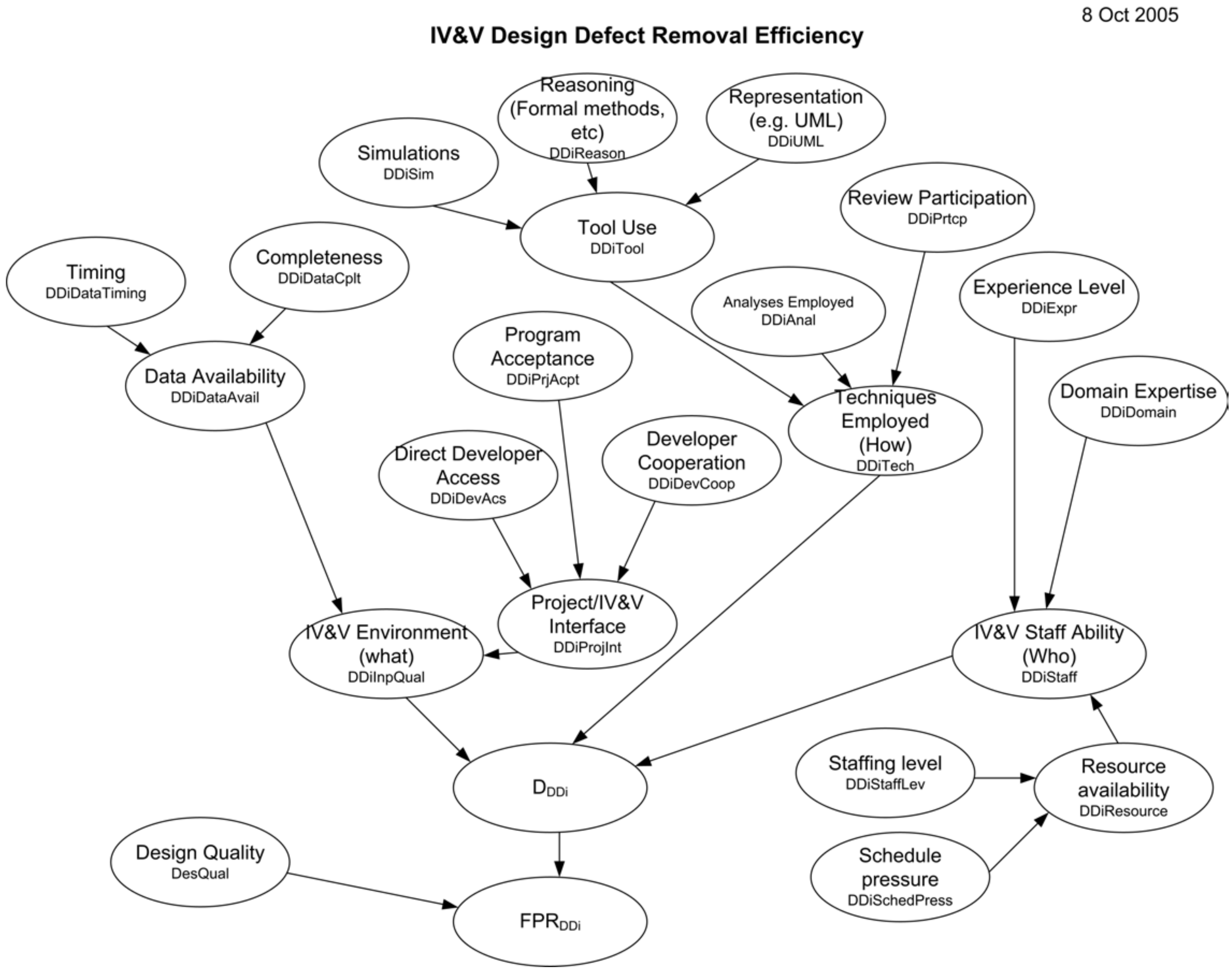

8 Oct 05
IV&V Requirements Defect Removal Efficiency
Reasoning (Formal methods, etc)
RRiReason
Representation (e.g. UML)
RRiUML
Simulations
RRiSim
Tool Use
RRiTool
Review Participation
RRiPrtcp
Timing
RRiDataTiming
Completeness
RRiDataCplt
Analyses Employed
RRiAnal
Experience Level
RRiExpr
Program Acceptance
RRiPrjAcpt
Data Availability
RRiDataAvail
Techniques Employed (How)
RRiTech
Domain Expertise
RRiDomain
Direct Developer Access
RRiDevAcs
Developer Cooperation
RRiDevCoop
Project/IV&V Interface
RRiProjInt
IV&V Environment (what)
RRiInpQual
IV&V Staff Ability (Who)
RRiStaff
D_{RRI}
Staffing level
RRiStaffLev
Resource availability
RRiResource
Requirements Quality
ReqQual
FPR_{RRI}
Schedule pressure
RRiSchedPress


8 Oct 2005
Design Defect Introduction
User Involvement
UserInvolve
User Systems Expertise
UserExper
Novelty of Problem or Approach
Novel
Quality of User Input
DesQualUserInput
Staff Level
DesDevStfLvl
Schedule pressure
DesDevSched
Budget margin
DesDevBudget
Process Effectiveness Actions
DesDevProcEff
Turnover
DesDevTrnOvr
Resource Availability
DesDevResAvail
External constraint pressure
DesExtCstr
Development Quality Organization
DesDevQualOrg
Staff Experience Level
DesDevExpr
Design Stability
DesStab
Domain Expertise
DesDevDomain
Process Definition, Product Standards, Quality Criteria (Req to Des seamlessness)
DevCMM
Process Adherence
DesProcAdhere
Requirements Quality
ReqQual
Problem Complexity
DesCplx
Problem Space (what)
DesProbSpc
Staff Ability (who)
DesDevStaff
Process Rigor (how)
DesProcRigor
Development Tools
DesDevTool
Design Quality
DesQual

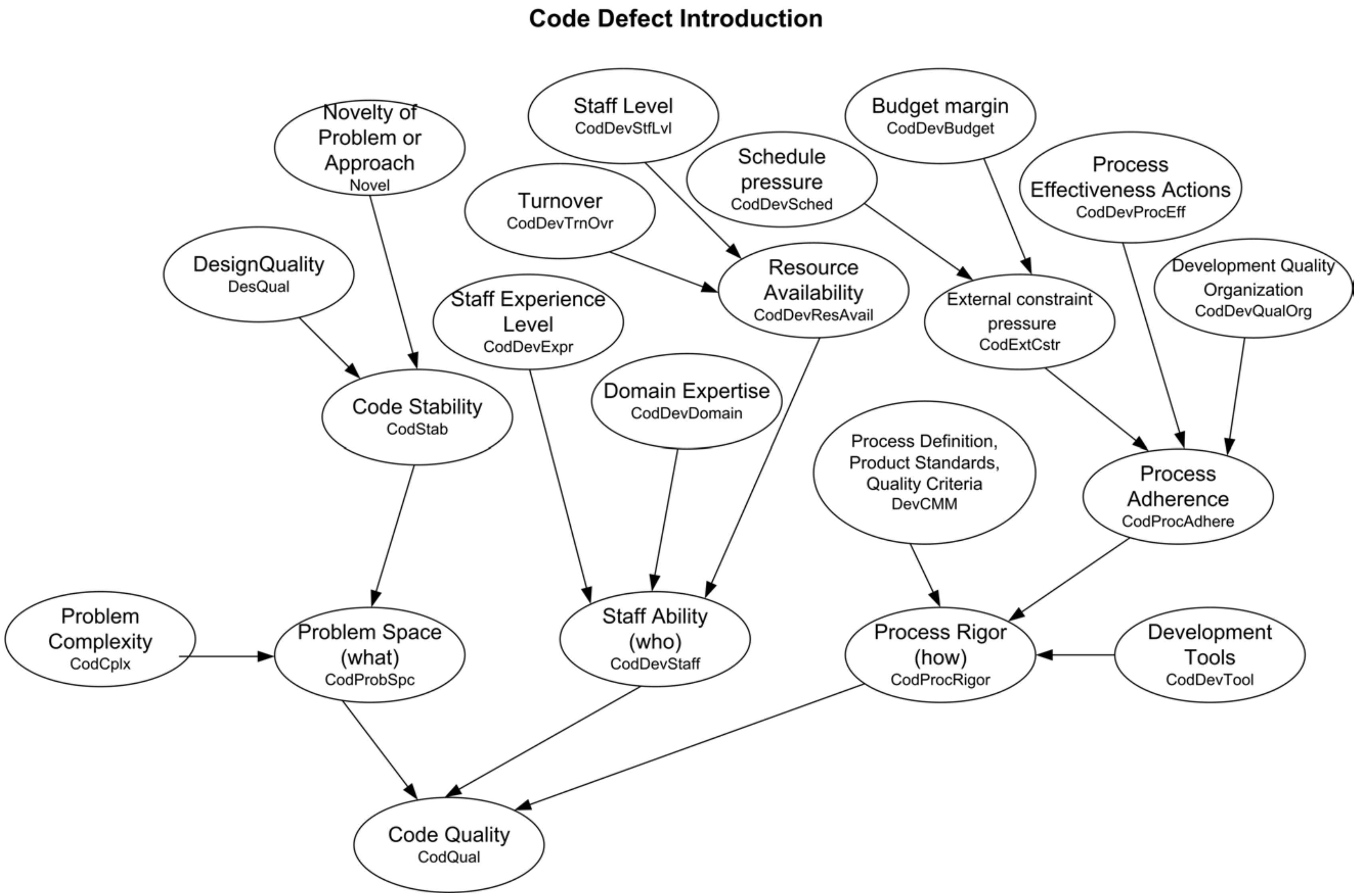
8 Oct 2005

Developer Design Defect Removal Efficiency



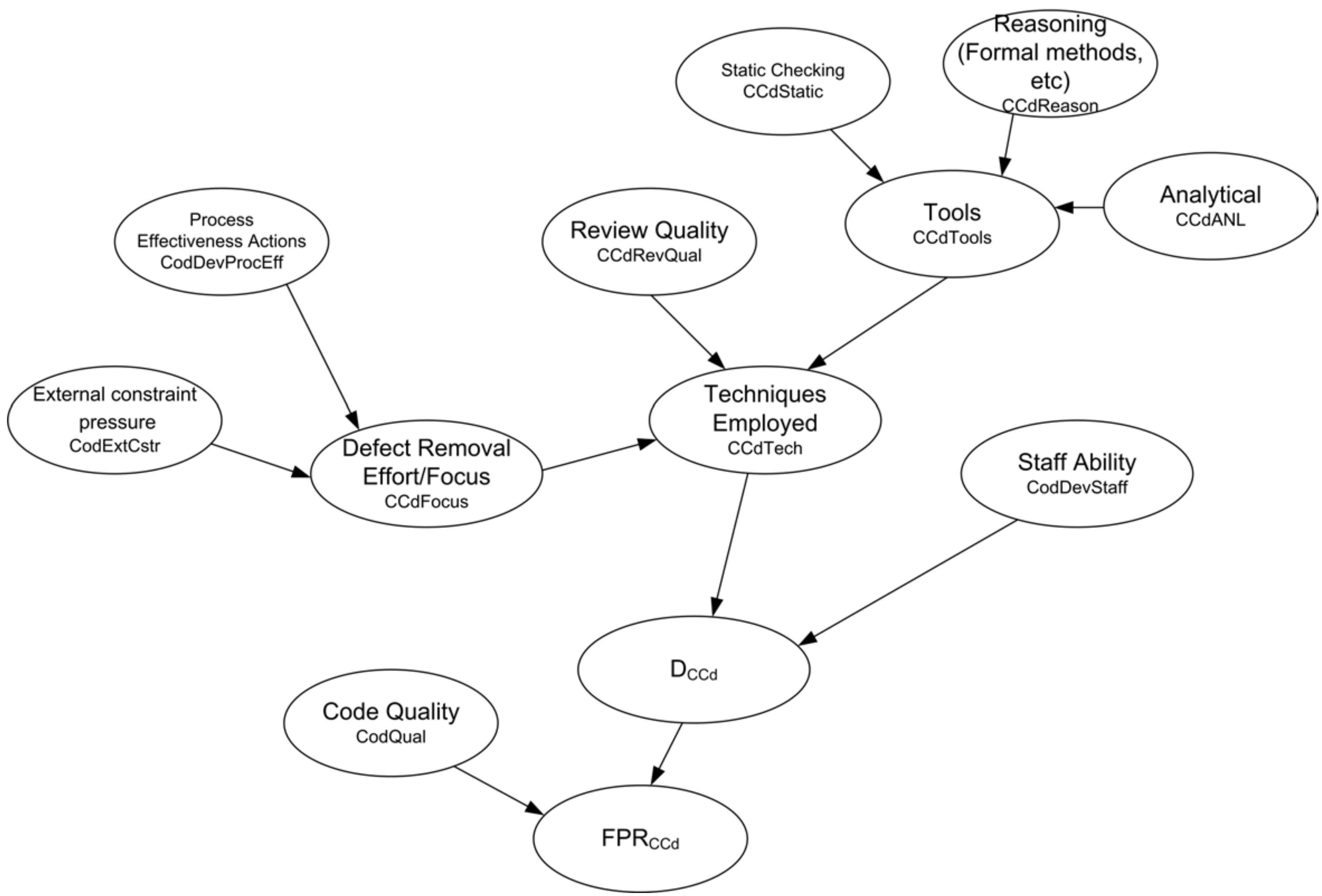

8 Oct 2005
IV&V Design Defect Removal Efficiency
Reasoning (Formal methods, etc) DDiReason
Representation (e.g. UML) DDiUML
Simulations DDiSim
Tool Use DDiTool
Review Participation DDiPrtcp
Timing DDiDataTiming
Completeness DDiDataCplt
Analyses Employed DDiAnal
Experience Level DDiExpr
Program Acceptance DDiPrjAcpt
Data Availability DDiDataAvail
Domain Expertise DDiDomain
Techniques Employed (How) DDiTech
Direct Developer Access DDiDevAcs
Developer Cooperation DDiDevCoop
Project/IV&V Interface DDiProjInt
IV&V Environment (what) DDiInpQual
IV&V Staff Ability (Who) DDiStaff
D DDi
Staffing level DDiStaffLev
Resource availability DDiResource
Design Quality DesQual
Schedule pressure DDiSchedPress
FPR DDi

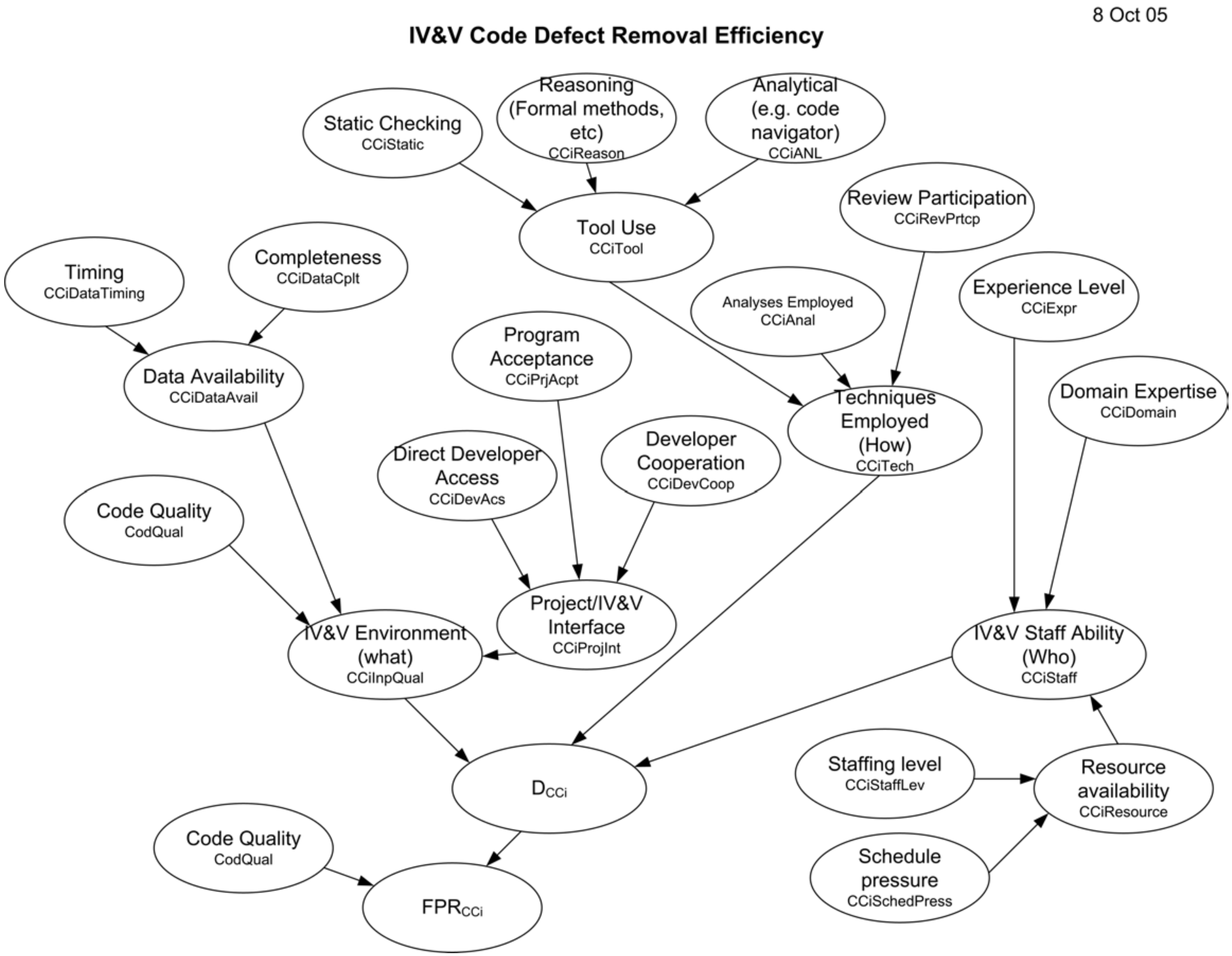
8 Oct 05

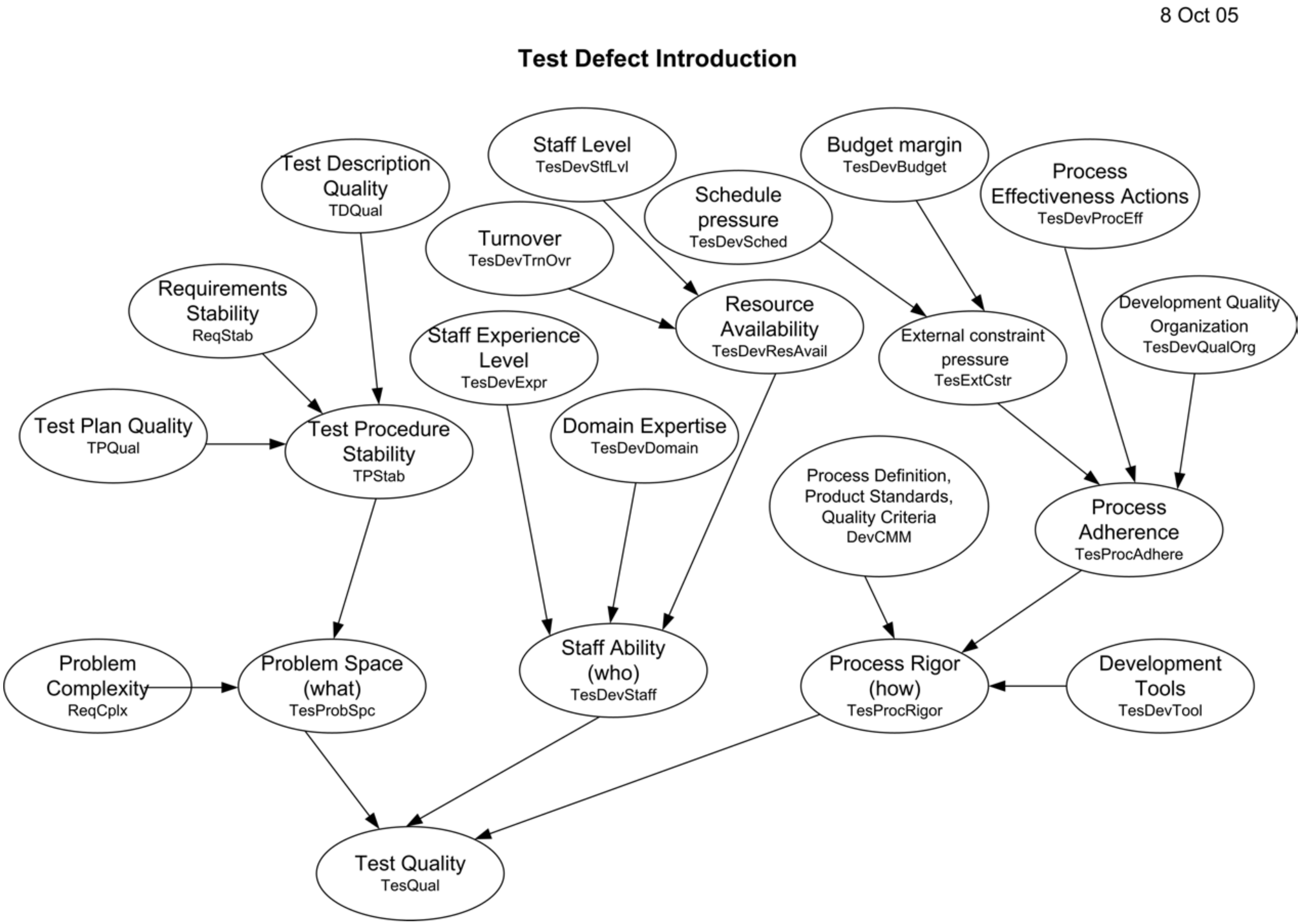

Code Defect Introduction
Novelty of Problem or Approach
Novel
Staff Level
CodDevStfLvl
Budget margin
CodDevBudget
Schedule pressure
CodDevSched
Process Effectiveness Actions
CodDevProcEff
Turnover
CodDevTrnOvr
DesignQuality
DesQual
Resource Availability
CodDevResAvail
Development Quality Organization
CodDevQualOrg
Staff Experience Level
CodDevExpr
External constraint pressure
CodExtCstr
Code Stability
CodStab
Domain Expertise
CodDevDomain
Process Definition, Product Standards, Quality Criteria
DevCMM
Process Adherence
CodProcAdhere
Problem Complexity
CodCplx
Problem Space (what)
CodProbSpc
Staff Ability (who)
CodDevStaff
Process Rigor (how)
CodProcRigor
Development Tools
CodDevTool
Code Quality
CodQual

8 Oct 05

Developer Code Defect Removal Efficiency

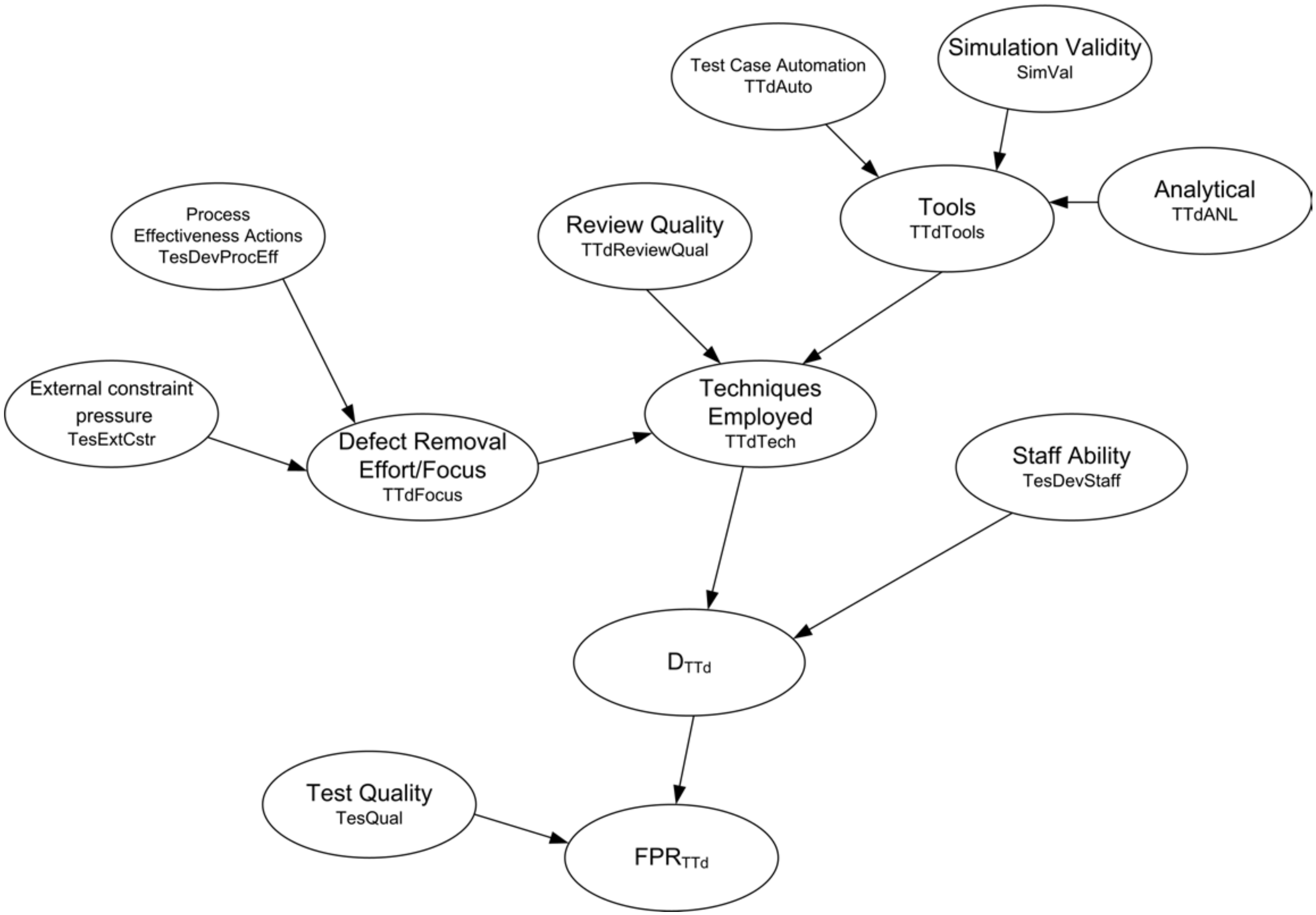


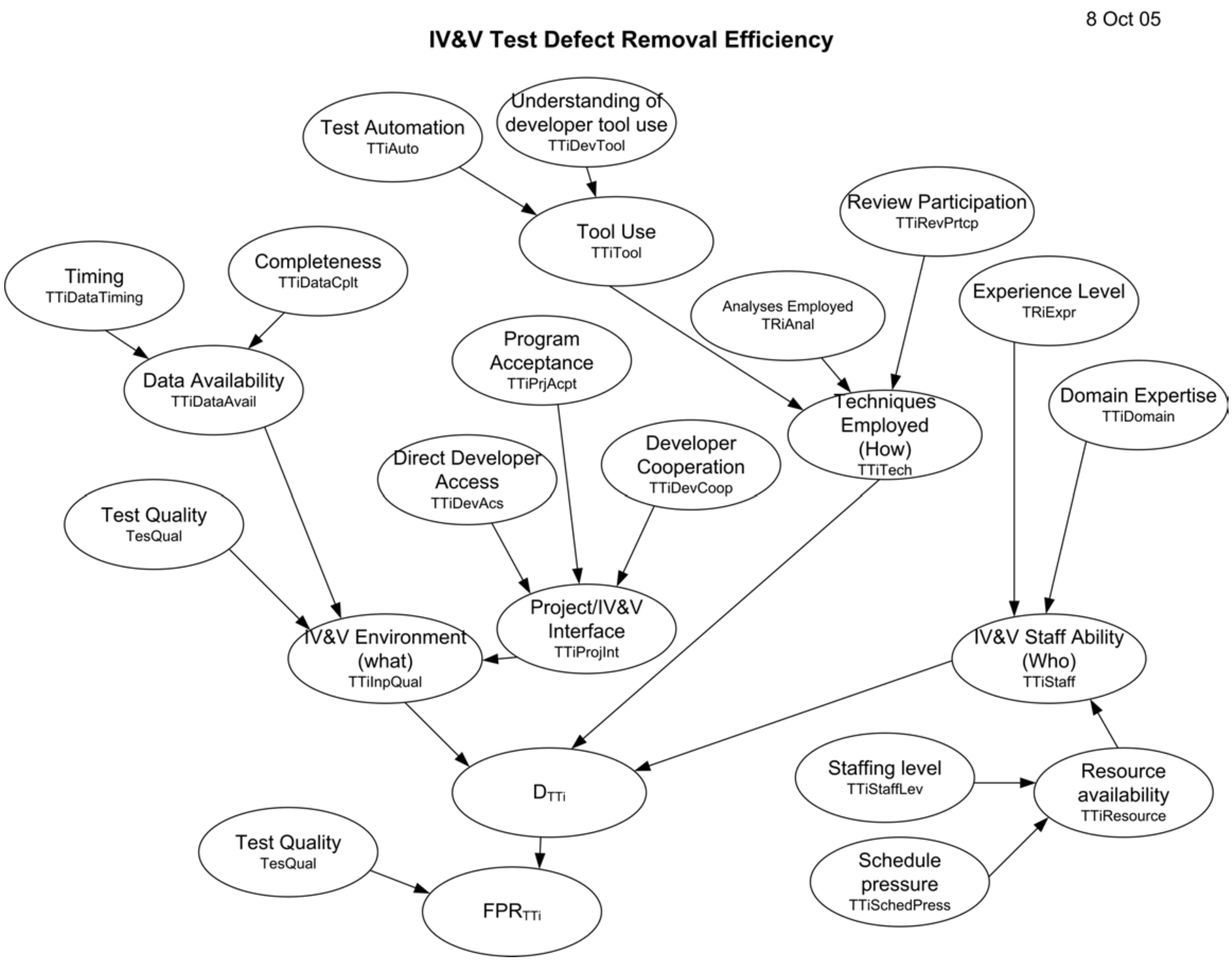

8 Oct 05
IV&V Code Defect Removal Efficiency
Static Checking
CCiStatic
Reasoning (Formal methods, etc)
CCiReason
Analytical (e.g. code navigator)
CCiANL
Tool Use
CCiTool
Review Participation
CCiRevPrtcp
Timing
CCiDataTiming
Completeness
CCiDataCplt
Analyses Employed
CCiAnal
Experience Level
CCiExpr
Program Acceptance
CCiPrjAcpt
Data Availability
CCiDataAvail
Techniques Employed (How)
CCiTech
Domain Expertise
CCiDomain
Direct Developer Access
CCiDevAcs
Developer Cooperation
CCiDevCoop
Code Quality
CodQual
Project/IV&V Interface
CCiProjInt
IV&V Environment (what)
CCiInpQual
IV&V Staff Ability (Who)
CCiStaff
D CCi
Staffing level
CCiStaffLev
Resource availability
CCiResource
Code Quality
CodQual
Schedule pressure
CCiSchedPress
FPR CCi


8 Oct 05
Test Defect Introduction
Staff Level
TesDevStfLvl
Budget margin
TesDevBudget
Test Description Quality
TDQual
Schedule pressure
TesDevSched
Process Effectiveness Actions
TesDevProcEff
Turnover
TesDevTrnOvr
Requirements Stability
ReqStab
Resource Availability
TesDevResAvail
Development Quality Organization
TesDevQualOrg
Staff Experience Level
TesDevExpr
External constraint pressure
TesExtCstr
Test Plan Quality
TPQual
Test Procedure Stability
TPStab
Domain Expertise
TesDevDomain
Process Definition, Product Standards, Quality Criteria
DevCMM
Process Adherence
TesProcAdhere
Problem Complexity
ReqCplx
Problem Space (what)
TesProbSpc
Staff Ability (who)
TesDevStaff
Process Rigor (how)
TesProcRigor
Development Tools
TesDevTool
Test Quality
TesQual

8 Oct 05

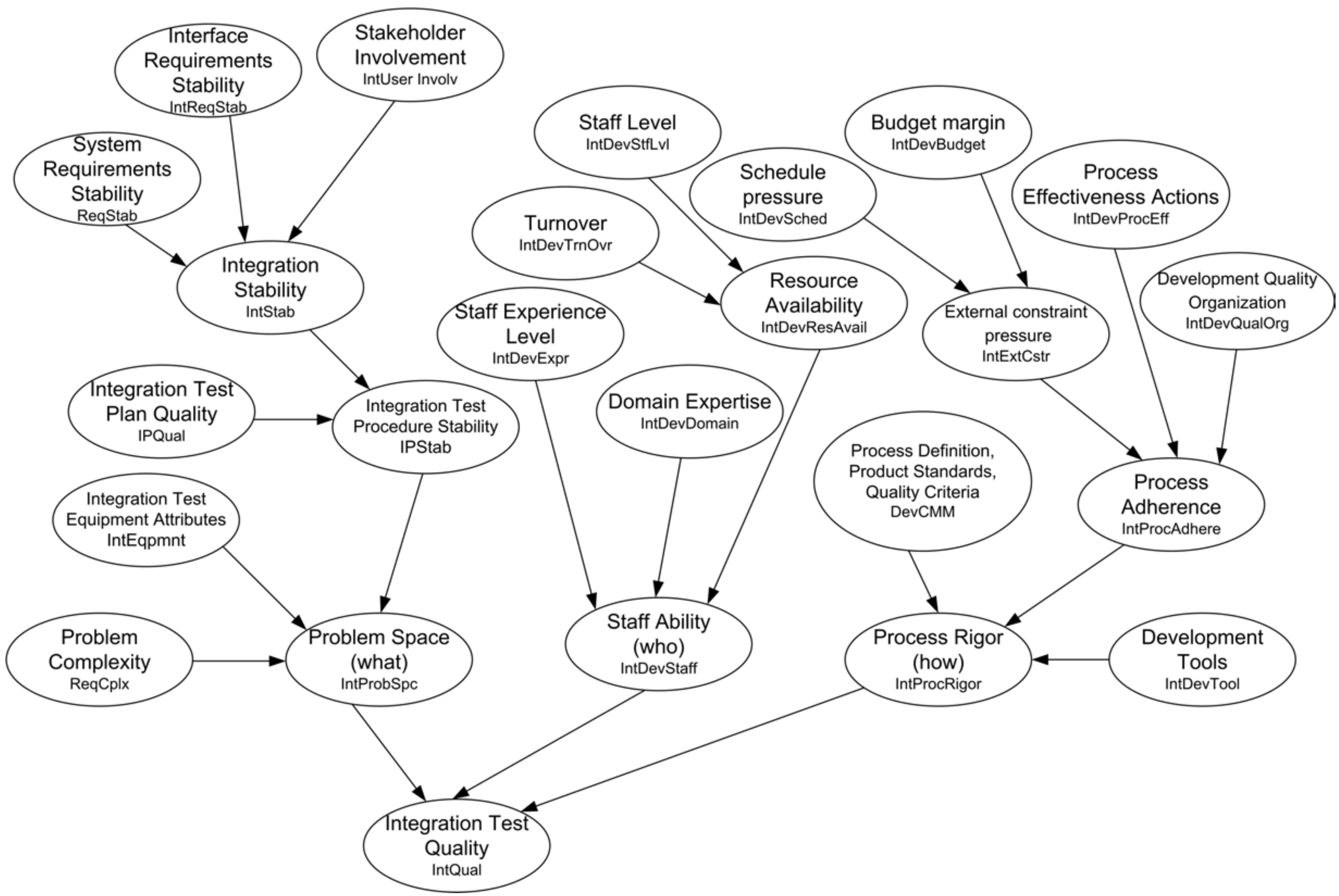
Developer Test Defect Removal Efficiency




8 Oct 05
IV&V Test Defect Removal Efficiency
Test Automation
TTiAuto
Understanding of developer tool use
TTiDevTool
Tool Use
TTiTool
Review Participation
TTiRevPrtcp
Timing
TTiDataTiming
Completeness
TTiDataCplt
Analyses Employed
TRiAnal
Experience Level
TRiExpr
Program Acceptance
TTiPrjAcpt
Data Availability
TTiDataAvail
Techniques Employed (How)
TTiTech
Domain Expertise
TTiDomain
Direct Developer Access
TTiDevAcs
Developer Cooperation
TTiDevCoop
Test Quality
TesQual
Project/IV&V Interface
TTiProjInt
IV&V Environment (what)
TTiInpQual
IV&V Staff Ability (Who)
TTiStaff
Staffing level
TTiStaffLev
Resource availability
TTiResource
D TTi
Test Quality
TesQual
Schedule pressure
TTiSchedPress
FPR TTi

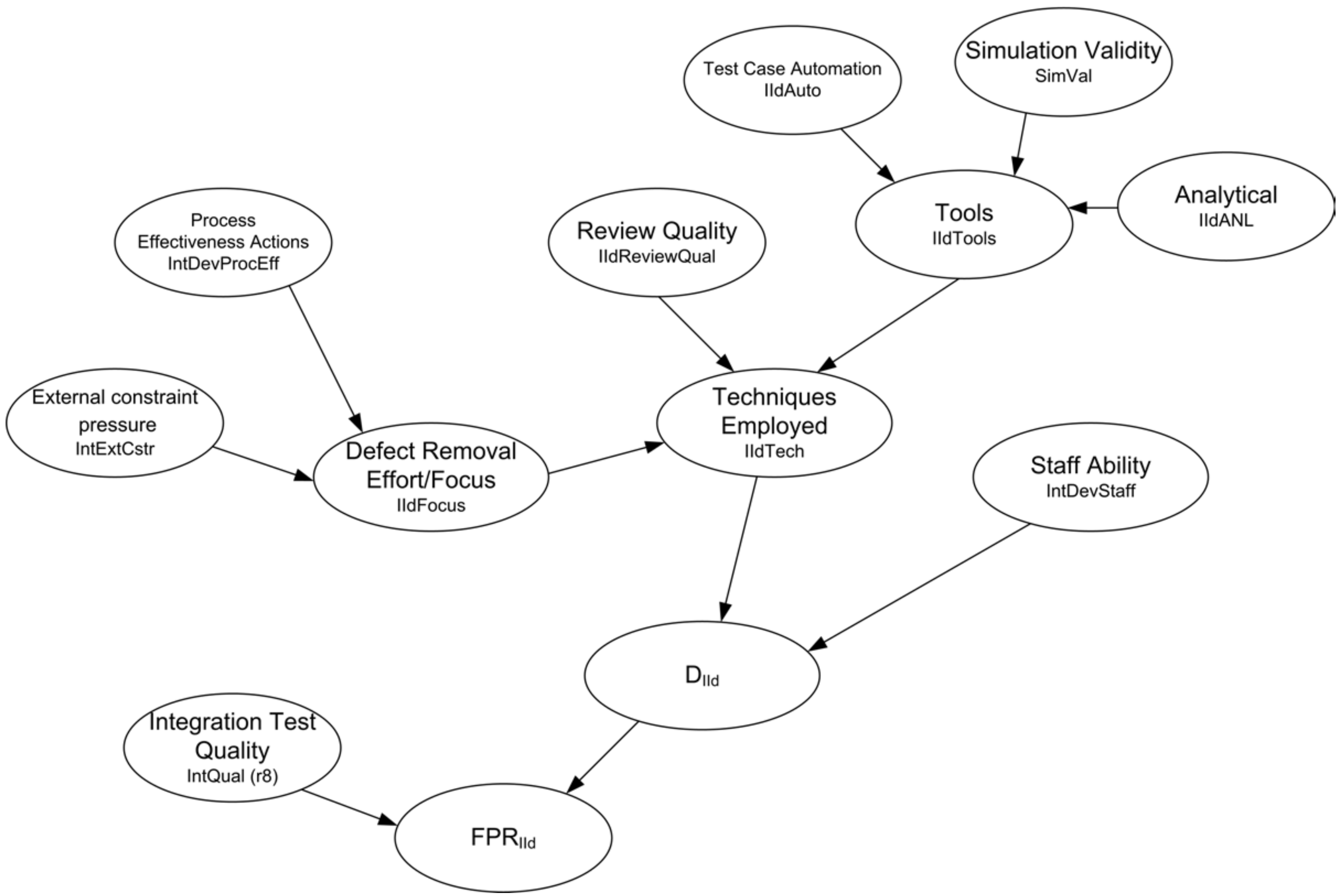
8 Oct 05

Integration Defect Introduction



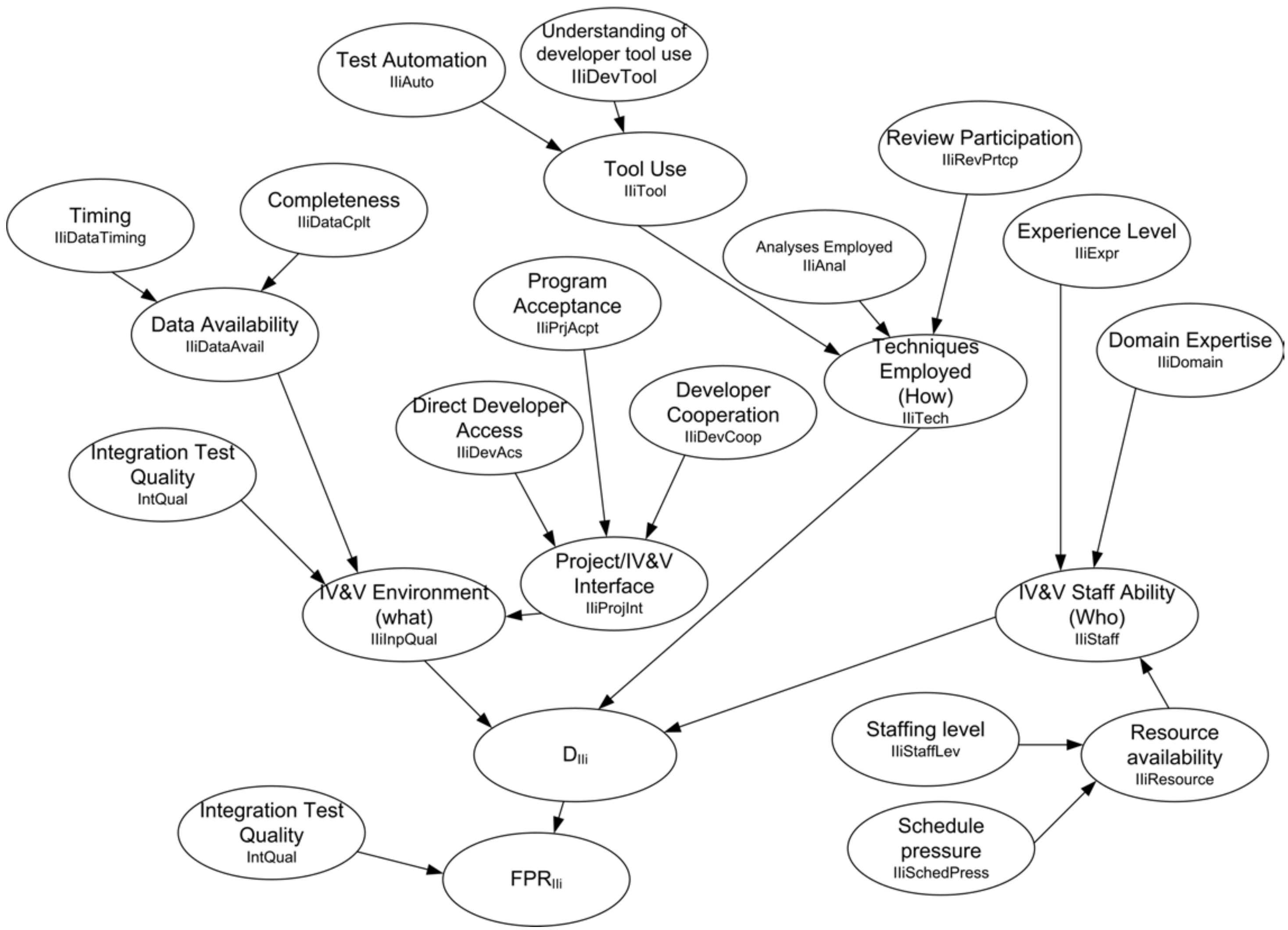
8 Oct 05

Developer Integration Defect Removal Efficiency



8 Oct 05

IV&V Integration Defect Removal Efficiency



Appendix B – BBN Input Definitions

B.1 Requirements Issue Subnet

Name	Characteristic Description
Requirements Defect Introduction	
User system expertise	Experience level of system users with similar (or the same) systems or solution approach. Maturity level of users (or representatives such as system engineering with equivalent knowledge) in understanding technical aspects of system to be implemented and the scenarios in which it will be used.
User involvement	Degree to which the system users are involved in the requirements definition process and the timeliness of that involvement. Note that a high score here requires involvement or representation (by system engineering, for example, with equivalent knowledge) of all key system users, not just system operators.
Heritage	Relative novelty (to the developer or user) of the application/mission or the solution approach. For example, entry GN&C for a new vehicle where all algorithms are adapted from Shuttle would be high heritage (score of 10) (provided the new mission is very similar to the Shuttle mission), completely new algorithms or new application would be low heritage (score of 1).
Quality of User Input	Effectiveness and timeliness of user involvement in assuring that requirements meet end user needs.
System documentation quality	Qualitative estimate of the quality (completeness, correctness, and consistency) of system documentation from which software requirements may be derived.
Problem complexity	Qualitative estimate of overall system/problem complexity. Related to technical difficulty to define, required interfaces (developer and system), and unity of users. Not correlated with code complexity metrics. Simple (10) to complex (1)
Requirements stability	How stable are requirements? The more stable, the fewer changes and the lower the risk of requirements errors.
Requirements Problem Space	Overall susceptibility of the problem space to introduction of requirements defects. Represents the difficulty, based on the complexity of the problem being solved and the quality and stability of documentation describing the problem, in deriving a correct set of requirements.
Dev staff experience level	Average experience level of development staff, not specific to the problem domain, but overall experience in software development for the domain type (e.g., real-time embedded flight, financial, ground, manned , etc).
Dev domain experience	Average experience of development staff in the specific application domain (e.g., laser guidance system, space telescope, crew rescue, etc). Consider all individual domains within the system (e.g., space telescope will require GN&C, optics, propulsion, system management, telemetry, etc).
Dev schedule pressure	How much margin is in the development schedule? Assessment of flexibility in end date. A higher number indicates developers have plenty of time to complete their work, a lower number indicates developers are consistently

	rushed to deliver products.
Dev budget margin	How tight is the development budget? Assessment of flexibility in cost growth.
External Constraint Pressure	Overall influence level of external constraints related to schedule and budget. How strong is the pressure to proceed without fix in spite of schedule or budget problems?
Process effectiveness actions	How much emphasis does development management place on quality (as opposed to productivity)? For example, does development management ensure that review action items are tracked, and encourage extra analysis of suspected problems? Assessment of effectiveness of process problem reports (e.g., a formal mechanism to document, correct and publish discrepancies in following the process), process improvement actions, activity of a board to assess effectiveness of process, etc. An assessment of the 'aliveness' of the developer process and attention to making it work to produce better products.
Dev quality organization	How effective is the embedded quality organization? This measure includes consideration of size, breadth and depth of capability applied to this project, and level of authority granted to the quality organization applied to this project.
Process Adherence	How well is the development staff likely to adhere to the documented requirements development process? Should be based on knowledge of schedule and budget constraints, level of activity related to process effectiveness, and the quality of organization enforcing the process.
Turnover	Experienced or historical rate of change of staff involved with requirements development. A higher number indicates little turnover, a lower number indicates a lot of turnover.
Staff level	Assessment of whether the quantity and distribution across domains of staff is sufficient for the problem space. An adequate staff level should receive a very high score. The worst staff level with respect to work required ever seen by the evaluator would receive a score of 1.
Resource Availability	Measure of the degree to which the size of the development staff is sufficient and stable in the terms of longevity on the project.
Staff Ability	Overall ability level of the requirements development staff, in terms of size, development experience, domain experience and turnover with respect to the problem at hand.
Process definition, product standards, quality criteria	This is an overall assessment of the effectiveness of the development process related to development of requirements. The assessment should include methods for requirements elicitation, coordination, documentation, and validation. It should correlate fairly well with CMM level, but includes an assessment of what is really happening in addition to what is documented.
Dev tools	Degree to which the developer uses tools in developing and analyzing requirements. Tools here include requirements management tools (DOORs, for example), traceability tools, simulations, process support, etc.
Process Rigor	How rigorous can the process be expected to be? Is the process well founded (related to CMM level), is it supported by a good set of tools, does the development organization pay

	attention to and follow the process?
Requirements quality	Overall relative measure of requirements quality, and hence, probable relative defect density.

Developer Requirements Defect Removal Efficiency	
Cost/schedule pressure	Similar to the cost and schedule pressure on requirements development. In some cases, the developer requirements inspection/checking process is more tightly constrained than requirements development.
Process effectiveness actions	Same as process effectiveness actions input for requirements development unless there is something unique about the group that reviews the products to find defects.
Stakeholder involvement	Degree to which users are involved in the requirements validation process. This is similar to but can be different from the user involvement for requirements development.
Defect Removal Effort/Focus	How much effort or focus does the developer apply to defect identification and removal during the requirements phase? Are users involved to make sure requirements are right, is there a rush to complete deliverables, is the defect removal process effective and adhered to?
Simulation tool use	Degree to which the development organization uses simulation to understand and validate requirements. This does not refer to simulations to verify requirements but rather to simulations of the conceptual operation of the system or particular subsystems to assure the right requirements are being specified. This could be a model of the expected environment and the system/subsystem reaction to it at an abstract rather than implementation level. Completeness of those simulations. Coverage of novel or complex areas.
Reasoning tool use	Degree of use of reasoning tools. By reasoning tools, we mean tools such as automation of formal methods, model checking, etc.
Representation tool use	Degree to which the development organization uses representational tools to automate and support requirements validation. The best example of representation tools is UML.
Tools effectiveness	Overall level and effectiveness of tool use in the requirements analysis process.
Review quality	Quality of the embedded review process. For example, do the developers do formal, detailed requirements reviews? Do they use entry and exit criteria, track and board issues, etc? Do they hold walkthroughs with broad scope support?
Techniques Employed effectiveness	Overall level and effectiveness of defect removal techniques used by the developer.

IV&V Requirements Defect Removal Efficiency	
Data timing	Relative measure of timeliness of data delivery by the developer to IV&V. Late data delivery places time restrictions on IV&V and can impede IV&V.
Data completeness	Measure of completeness of data submitted to IV&V. For example, if important information is withheld due to proprietary concerns, that will impede IV&V. Another example is that documents are delivered with sections incomplete or many TBDs or document does not have content expected for the point in lifecycle at which it was delivered.
Data availability	Likelihood that sufficient and timely input data will be available to IV&V when it is needed.
IV&V environment	Overall operating environment given to IV&V. Are artifacts on time and do they contain what is needed for efficient evaluation, are issues considered fairly and in a timely manner?
Direct dev access	How much access does IV&V have to the developers? If IV&V has to work through several bureaucratic levels to get information from the developers or discuss issues or risks, that will decrease IV&V effectiveness.
Project acceptance	How well does the project accept the IV&V participation? A high score indicates the project participants exhibit a belief that use of IV&V will lead to higher mission success probability. Project management has issued directives exhibiting the right intent in dealing with the IV&V participation.
Dev cooperation	How cooperative are developers, in general, in responding to IV&V requests and suggestions? Do all IV&V issues get immediate attention, or does the development organization tend to ignore or avoid dealing with IV&V concerns? Developers exhibit attention to IV&V concerns and timely/meaningful response. While project acceptance is having the right intent, cooperation is doing the right thing.
Project/IV&V interface	How efficient is the interface between the developer and IV&V considering access, cooperation and acceptance.
IV&V experience level	Average experience level of IV&V staff. This is a measure of how much experience the IV&V staff has in the area of IV&V and related activities. This does not consider domain experience level which is considered in another input.
IV&V domain experience/expertise level	Degree of experience of IV&V staff with the application domain. Consider all individual domains within the system (e.g., GN&C, power, C&DH, ECLSS, terrain mobility, thermal, etc). This item should evaluate the extent of applicable domain knowledge within the IV&V staff.
IV&V Staffing level	How appropriate is the size of the IV&V staff to the analysis tasks that need to be performed based on the CARA results? Too few (or too many) would lower the rating. An adequate staff level should receive a very high score. Descoping the tasks from the CARA results to match a low staff level would get a low score.
Schedule pressure	How much schedule pressure does IV&V face. This factor could be correlated with data timing, but not necessarily. A low score indicates there was heavy schedule pressure. A high score indicates there was little schedule pressure.
Resource availability	Availability of all human resources needed to perform IV&V. A

	sufficient set of personnel is available to perform the IV&V activities including consideration of the schedule pressure from the project (e.g., very short turnaround of document review expected), size of staff and personnel turnover.
IV&V Staff ability	Overall ability of IV&V staff to perform the IV&V tasks defined by the CARA analysis.
Simulation tool use	Degree to which the IV&V organization uses simulation to understand and validate requirements. This does not refer to simulations to verify requirements but rather to simulations of the conceptual operation of the system or particular subsystems to assure the right requirements are being specified. This could be a model of the expected environment and the system/subsystem reaction to it at an abstract rather than implementation level. Completeness of those simulations. Coverage of novel or complex areas.
Reasoning tool use	Degree of use of reasoning tools by the IV&V organization. By reasoning tools, we mean tools such as automation of formal methods, model checking, traceability, etc.
Representation tool use	Degree to which the IV&V organization uses representational tools to automate and support requirements validation. The best example of representation tools is UML.
IV&V tool effectiveness	An assessment of effectiveness of all tools employed to support requirements validation including simulations, formal method support, UML, requirements trace etc.
Analyses employed	An assessment of the effectiveness of the types of analyses planned such as scenario analyses, requirements reading, comparative analysis, formal method
Developer review participation	Assessment of effectiveness of plans for participating in developer milestone reviews, inspections, walkthroughs etc. Assessment would include timing in which IV&V enters the project (SRR, SDR SSR, etc) and its impact on effectiveness.
IV&V techniques employed effectiveness	Overall level and effectiveness of defect removal techniques used by IV&V

B.2 Design Issue Subnet

Name	Characteristic Description
	Design Defect Introduction
User system expertise	Experience level of system users with similar (or the same) systems or solution approach. Maturity level of users (or representatives such as system engineering with equivalent knowledge) in understanding technical aspects of system to be implemented and the scenarios in which it will be used.
User involvement	Degree to which the system users are involved in the requirements definition process and the timeliness of that involvement. Note that a high score here requires involvement or representation (by system engineering, for example, provided they have equivalent knowledge) of all key system users, not just system operators.
Heritage	Relative novelty (to the developer or user) of the application/mission or the solution approach. For example, entry GN&C for a new vehicle where all algorithms are adapted from Shuttle would be high heritage (score of 10) (provided the new mission is very similar to the Shuttle mission), completely new algorithms or new application would be low heritage (score of 1).
Quality of User Input	Effectiveness and timeliness of user involvement in assuring that design meets end user needs.
Problem complexity	Qualitative estimate of overall system/problem complexity. Related to technical difficulty to define system, required interfaces (among the developer and system), and unity of users. Not correlated with code complexity metrics. Simple (10) to complex (1
Design stability	How stable is the design? This is driven by the heritage of the approach, support from the users, requirements stability, and the proneness of the design group to make errors. The more stable, the fewer changes and the lower the risk of introduction of new errors.
Design Problem Space	Overall susceptibility of the problem space to introduction of design defects. Represents the difficulty, based on the complexity of the problem being solved and the quality and stability of documentation describing the problem, in deriving a correct design.
Dev staff experience level	Average experience level of development staff, not specific to the problem domain, but overall experience in software development for the domain type (e.g., real-time embedded flight, financial, ground, manned , etc). Includes staff experience with the language chosen for implementation and the operating system in use. Also includes experience with the processor to be used, and the hardware to be interfaced with
Dev domain experience	Average experience of development staff in the specific application domain (e.g., laser guidance system, space telescope, crew rescue, etc). Consider all individual domains within the system (e.g., space telescope will require GN&C, optics, propulsion, system management, telemetry, etc).
Dev schedule pressure	How much margin is in the development schedule? Assessment of flexibility in end date. A higher number indicates developers have plenty of time to complete their work, a lower number

	indicates developers are consistently rushed to deliver products.
Dev budget margin	How tight is the development budget? Assessment of flexibility in cost growth.
External Constraint Pressure	Overall influence level of external constraints related to schedule and budget. How strong is the pressure to proceed without fix in spite of schedule or budget problems?
Process effectiveness actions	How much emphasis does development management place on quality (as opposed to productivity)? For example, does development management ensure that review action items are tracked, and encourage extra analysis of suspected problems? Assessment of effectiveness of process problem reports (e.g., a formal mechanism to document, correct and publish discrepancies in following the process), process improvement actions, activity of a board to assess effectiveness of process, etc. An assessment of the 'aliveness' of the developer process and attention to making it work to produce better products.
Dev quality organization	How effective is the embedded quality organization? This measure includes consideration of size, breadth and depth of capability applied to this project, and level of authority granted to the quality organization applied to this project
Process Adherence	How well is the development staff likely to adhere to the documented design development process? Should be based on knowledge of schedule and budget constraints, level of activity related to process effectiveness, and the quality of organization enforcing the process.
Turnover	Experienced or historical rate of change of staff involved with design development. A higher number indicates little turnover, a lower number indicates a lot of turnover
Staff level	Assessment of whether the quantity and distribution across domains of staff is sufficient for the problem space. An adequate staff level should receive a very high score (not average). The worst staff level with respect to work required ever seen by the evaluator would receive a score of 1.
Resource Availability	Measure of the degree to which the size of the development staff is sufficient and stable in terms of longevity on the project
Staff Ability	Overall ability level of the design development staff, in terms of size, development experience, domain experience and turnover with respect to the problem at hand
Process definition, product standards, quality criteria	This is an overall assessment of the effectiveness of the development process related to development of design. The assessment should include methods for design derivation, coordination, documentation, and validation. It should correlate fairly well with CMM level, but includes an assessment of what is really happening in addition to what is documented.
Dev tools	Degree to which the developer uses tools in developing and analyzing design. Tools here include formal method support tools (Stateflow, for example), traceability tools, simulations, process support, etc. The assessment should include evaluation of tool support to assure conformance of design to requirements (i.e., do tools support a seamless transition to design from requirements or are the design support tools completely independent of requirements support tools)
Process Rigor	How rigorous can the process be expected to be? Is the process well founded (related to CMM level), is it supported by a

	good set of tools, does the development organization pay attention to and follow the process?
Design quality	Overall relative measure of design quality, and hence, probable relative defect density.

Developer Design Defect Removal Efficiency	
External Constraint Pressure	Similar to the cost and schedule pressure on design development. In some cases, the developer design inspection/checking process is more tightly constrained than design development.
Process effectiveness actions	Same as process effectiveness actions input for design development unless there is something unique about the group that reviews the products to find defects.
Stakeholder involvement	Degree to which users are involved in the design validation process. This is similar to but can be different from the user involvement for design development
Defect Removal Effort/Focus	How much effort or focus does the developer apply to defect identification and removal during the design phase? Are users involved to make sure design is right, is there a rush to complete deliverables, is the defect removal process effective and adhered to?
Simulation tool use	Degree to which the development organization uses simulation to understand and validate design. This does not refer to simulations to verify requirements but rather to simulations of the conceptual operation of the system or particular subsystems to assure the right design is being specified. This could be a model of the expected environment and the system/subsystem reaction to it at an abstract rather than implementation level. Completeness of those simulations. Coverage of novel or complex areas.
Reasoning tool use	Degree of use of reasoning tools. By reasoning tools, we mean tools such as automation of formal methods, model checking, etc. Extent of executability of design. Ability of tools to point to defects in design characteristics such as sequencing, timing, homogeneity, etc.
Representation tool use	Degree to which the development organization uses representational tools to automate and support design validation. The best example of representation tools is UML
Tools effectiveness	Overall level and effectiveness of tool use in the design analysis process for removal of defects.
Review quality	Quality of the embedded review process. For example, do the developers do formal, detailed design reviews? Do they use entry and exit criteria, track and board issues, etc? Do they hold walkthroughs with broad scope support?
Techniques Employed effectiveness	Overall level and effectiveness of defect removal techniques used by the developer

IV&V Design Defect Removal Efficiency	
Data timing	Relative measure of timeliness of data delivery by the developer to IV&V. Late data delivery places time restrictions on IV&V and can impede IV&V.
Data completeness	Measure of completeness of data submitted to IV&V. For example, if important information is withheld due to proprietary concerns, that will impede IV&V. Also, documents delivered with sections incomplete or many TBDs or lacking content expected for the point in lifecycle at which it was delivered.
Data availability	Likelihood that sufficient and timely input data will be available to IV&V when it is needed.
IV&V environment	Overall operating environment given to IV&V. Are artifacts on time and do they contain what is needed for efficient evaluation, are issues considered fairly and in a timely manner?
Direct dev access	How much access does IV&V have to the developers? If IV&V has to work through several bureaucratic levels to get information from the developers or discuss issues or risks, that will decrease IV&V effectiveness.
Project acceptance	How well does the project accept the IV&V participation? A high score indicates the project participants exhibit a belief that use of IV&V will lead to higher mission success probability. Project management has issued directives exhibiting the right intent in dealing with the IV&V participation.
Dev cooperation	How cooperative are developers, in general, in responding to IV&V requests and suggestions? Do all IV&V issues get immediate attention, or does the development organization tend to ignore or avoid dealing with IV&V concerns? Developers exhibit attention to IV&V concerns and timely/meaningful response. While project acceptance is having the right intent, cooperation is doing the right thing.
Project/IV&V interface	How efficient is the interface between the developer and IV&V considering access, cooperation and acceptance.
IV&V experience level	Average experience level of IV&V staff. This is a measure of how much experience the IV&V staff has in the area of IV&V and related activities. This does not consider domain experience level which is considered in another input. This considers experience in implementation language, operating system in use, processing platform and hardware interfaces
IV&V domain experience/expertise level	Degree of IV&V staff experience with the application domain. Consider all individual domains within the system (GN&C, power, C&DH, ECLSS, terrain mobility, thermal, etc). This item should evaluate the extent of applicable domain knowledge within the IV&V staff.
IV&V Staffing level	How appropriate is IV&V staff size to the needed analysis tasks, based on the CARA results? Too few (or too many) would lower the rating. An adequate staff level should receive a very high score (not average). Descoping the tasks from the CARA results to match a low staff level would get a low score
Schedule pressure	How much schedule pressure does IV&V face. This factor could be correlated with data timing, but not necessarily. A low score indicates there was heavy schedule pressure. A high score indicates there was little schedule pressure
Resource availability	Availability of all human resources needed to perform IV&V. A

	sufficient set of personnel is available to perform the IV&V activities including consideration of the schedule pressure from the project (e.g., very short turnaround of document review expected), size of staff and personnel turnover.
IV&V Staff ability	Overall ability of IV&V staff to perform the IV&V tasks defined by the CARA analysis.
Simulation tool use	Degree to which the IV&V organization uses simulation to understand and validate design. This does not refer to simulations to verify requirements but rather to simulations of the conceptual operation of the system or particular subsystems to assure the right design is being specified. This could be a model of the expected environment and the system/subsystem reaction to it at an abstract rather than implementation level. Completeness of those simulations. Coverage of novel or complex areas.
Reasoning tool use	Degree of use of reasoning tools by the IV&V organization. By reasoning tools, we mean tools such as automation of formal methods, model checking, traceability, etc.
Representation tool use	Degree to which the IV&V organization uses representational tools to automate and support design validation. The best example of representation tools is UML or a combination of design language and graphics such as AADL. This represents the degree to which IV&V develops their own design representation for analysis and the degree to which IV&V has the tools to understand the developer's representation of the design. For the case in which design documentation is very poor and code navigation tools are used to support understanding the design, those tools may be scored here.
IV&V tool effectiveness	An assessment of effectiveness of all tools employed to support design validation including simulations, formal method support, UML, requirements trace etc.
Analyses employed	An assessment of the effectiveness of the types of analyses planned such as scenario analyses, design reading, comparative analysis, and formal method.
Developer review participation	Assessment of effectiveness of plans for participating in developer milestone reviews, inspections, walkthroughs etc. Assessment would include timing in which IV&V enters the project (SRR, SDR SSR, etc) and its impact on effectiveness.
IV&V techniques employed effectiveness	Overall level and effectiveness of defect removal techniques used by IV&V

B.3 Code Issue Subnet

Name	Characteristic Description
	Code Defect Introduction
Design Quality	Quantitative estimate of design quality coming from the design defect introduction subnet
Heritage	Relative novelty (to the developer or user) of the application/mission or the solution approach. For example, entry GN&C for a new vehicle where all algorithms are adapted from Shuttle would be high heritage (score of 10) (provided the new mission is very similar to the Shuttle mission), completely new algorithms or new application would be low heritage (score of 1).
Problem complexity	Qualitative estimate of overall system/problem complexity. Related to technical difficulty to define system, required interfaces (among the developer and system), and unity of users. Not correlated with code complexity metrics. Simple (10) to complex (1).
Code stability	How stable is the code? This is driven by the heritage of the approach, design quality, and the proneness of the code group to make errors. The more stable, the fewer changes and the lower the risk of introduction of new errors.
Code Problem Space	Overall susceptibility of the problem space to introduction of code defects. Represents the difficulty, based on the complexity of the problem being solved, the algorithms chosen, and the quality and stability of documentation describing the problem, in deriving a correct implementation.
Dev staff experience level	Average experience level of development staff, not specific to the problem domain, but overall experience in software development for the domain type (e.g., real-time embedded flight, financial, ground, manned, etc). Includes staff experience with the language chosen for implementation and the operating system in use. Also includes experience with the processor to be used, and the hardware to be interfaced with
Dev domain experience	Average experience of development staff in the specific application domain (e.g., laser guidance system, space telescope, crew rescue, etc). Consider all individual domains within the system (e.g., space telescope will require GN&C, optics, propulsion, system management, telemetry, etc)
Dev schedule pressure	How much margin is in the development schedule? Assessment of flexibility in end date. A higher number indicates developers have plenty of time to complete their work, a lower number indicates developers are consistently rushed to deliver products.
Dev budget margin	How tight is the development budget? Assessment of flexibility in cost growth.
External Constraint Pressure	Overall influence level of external constraints related to schedule and budget. How strong is the pressure to proceed without fix in spite of schedule or budget problems?
Process effectiveness actions	How much emphasis does development management place on quality (as opposed to productivity)? For example, does development management ensure that review action items are tracked, and encourage extra analysis of suspected problems? Assessment of effectiveness of process problem reports (e.g., a formal mechanism to document, correct and publish

	discrepancies in following the process), process improvement actions, activity of a board to assess effectiveness of process, etc. An assessment of the 'aliveness' of the developer process and attention to making it work to produce better products.
Dev quality organization	How effective is the embedded quality organization? This measure includes consideration of size, breadth and depth of capability applied to this project, and level of authority granted to the quality organization applied to this project.
Process Adherence	How well is the development staff likely to adhere to the documented code development process? Should be based on knowledge of schedule and budget constraints, level of activity related to process effectiveness, and the quality of organization enforcing the process.
Turnover	Experienced or historical rate of change of staff involved with code development. A higher number indicates little turnover, a lower number indicates a lot of turnover.
Staff level	Assessment of whether the quantity and distribution across domains of staff is sufficient for the problem space. An adequate staff level should receive a very high score (not average). The worst staff level with respect to work required ever seen by the evaluator would receive a score of 1.
Resource Availability	Measure of the degree to which the size of the development staff is sufficient and stable in terms of longevity on the project.
Staff Ability	Overall ability level of the code development staff, in terms of size, development experience, domain experience and turnover with respect to the problem at hand
Process definition, product standards, quality criteria	This is an overall assessment of the effectiveness of the development process related to development of code. The assessment should include methods for code development and debug, unit test, integration test, commenting, and coding standards. It should correlate fairly well with CMM level, but includes an assessment of what is really happening in addition to what is documented.
Dev tools	Degree to which the developer uses tools in developing and analyzing code. Tools here include formal method support tools, traceability tools, code navigation, process support, static and dynamic code analysis tools, etc. The assessment should include evaluation of tool support to assure conformance of code to design and requirements (i.e., do tools support a seamless transition to code from design or are the code support tools completely independent of design and requirements support tools).
Process Rigor	How rigorous can the process be expected to be? Is the process well founded (related to CMM level), is it supported by a good set of tools, does the development organization pay attention to and follow the process?
Code quality	Overall relative measure of code quality, and hence, probable relative defect density.

Developer Code Defect Removal Efficiency	
External Constraint Pressure	Similar to the cost and schedule pressure on code development. In some cases, the developer code inspection/checking process is more tightly constrained than code development.
Process effectiveness actions	Same as process effectiveness actions input for code development unless there is something unique about the group that reviews the products to find defects
Defect Removal Effort/Focus	How much effort or focus does the developer apply to defect identification and removal during the code phase? Are users involved to make sure code is right, is there a rush to complete deliverables, is the defect removal process effective and adhered to?
Static checkers	Degree and effectiveness of use of automated defect locator tools such as Lint, Coverity, or Polyspace. These are tools that do a static analysis of source code and identify potentially erroneous constructs.
Reasoning tool use	Degree of use of reasoning tools. By reasoning tools, we mean tools such as automation of formal methods, model checking, etc. An example would be SPIN or Rational Rose that use finite state machine representation in Promela or UML to provide an executable model. Ability of tools to point to defects in code characteristics such as sequencing, timing, homogeneity, etc.
Analytical tool use	Degree to which the development organization uses analytical tools to understand and write/modify code. These include tools such as target system debuggers, code navigators (e.g., Understand), or flow charters
Tools effectiveness	Overall level and effectiveness of tool use in the code analysis process for removal of defects.
Review quality	Quality of the embedded review process. For example, do the developers do formal, detailed code reviews? Do they use entry and exit criteria, track and board issues, etc? Do they hold walkthroughs with broad scope support?
Techniques Employed effectiveness	Overall level and effectiveness of defect removal techniques used by the developer.

IV&V Code Defect Removal Efficiency	
Data timing	Relative measure of timeliness of data delivery by the developer to IV&V. Late data delivery places time restrictions on IV&V and can impede IV&V.
Data completeness	Measure of completeness of data submitted to IV&V. For example, if important information is withheld due to proprietary concerns, that will impede IV&V. Another example is that source is delivered with sections incomplete or many TBDs or source does not have content expected for the point in lifecycle at which it was delivered.
Data availability	Likelihood that sufficient and timely input data will be available to IV&V when it is needed.
IV&V environment	Overall operating environment given to IV&V. Are artifacts delivered on time and do they contain what is needed for efficient evaluation, are issues considered fairly and in a timely manner?
Direct dev access	How much access does IV&V have to the developers? If IV&V has to work through several bureaucratic levels to get information from the developers or discuss issues or risks, that will decrease IV&V effectiveness.
Project acceptance	How well does the project accept the IV&V participation? A high score indicates the project participants exhibit a belief that use of IV&V will lead to higher mission success probability. Project management has issued directives exhibiting the right intent in dealing with the IV&V participation.
Dev cooperation	How cooperative are developers, in general, in responding to IV&V requests and suggestions? Do all IV&V issues get immediate attention, or does the development organization tend to ignore or avoid dealing with IV&V concerns? Developers exhibit attention to IV&V concerns and timely/meaningful response. While project acceptance is having the right intent, cooperation is doing the right thing.
Project/IV&V interface	Average experience level of IV&V staff. This is a measure of how much experience the IV&V staff has in the area of IV&V and related activities. This does not consider domain experience level which is considered in another input. This considers experience in implementation language, operating system in use, processing platform and hardware interfaces.
IV&V experience level	Average experience level of IV&V staff. This is a measure of how much experience the IV&V staff has in the area of IV&V and related activities. This does not consider domain experience level which is considered in another input. This considers experience in implementation language, operating system in use, processing platform and hardware interfaces.
IV&V domain experience/expertise level	Degree of experience of IV&V staff with the application domain. Consider all individual domains within the system (e.g., GN&C, power, C&DH, ECLSS, terrain mobility, thermal, etc). This item should evaluate the extent of applicable domain knowledge within the IV&V staff.
IV&V Staffing level	How appropriate is the size of the IV&V staff to the analysis tasks that need to be performed based on the CARA results? Too few (or too many) would lower the rating. An adequate staff level should receive a very high score (not average). Descoping

	the tasks from the CARA results to match a low staff level would get a low score.
Schedule pressure	How much schedule pressure does IV&V face. This factor could be correlated with data timing, but not necessarily. A low score indicates there was heavy schedule pressure. A high score indicates there was little schedule pressure.
Resource availability	Availability of all human resources needed to perform IV&V. A sufficient set of personnel is available to perform the IV&V activities including consideration of the schedule pressure from the project (e.g., very short turnaround of document review expected), size of staff and personnel turnover.
IV&V Staff ability	Overall ability of IV&V staff to perform the IV&V tasks defined by the CARA analysis.
Static checkers	Degree and effectiveness of use of automated defect locator tools such as Lint, Coverity, or Polyspace. These are tools that do a static analysis of source code and identify potentially erroneous constructs.
Reasoning tool use	Degree of use of reasoning tools. By reasoning tools, we mean tools such as automation of formal methods, model checking, etc. An example would be SPIN or Rational Rose that use finite state machine representation in Promela or UML to provide an executable model. Ability of tools to point to defects in code characteristics such as sequencing, timing, homogeneity, etc.
Analytical tool use	Degree to which the IV&V organization uses analytical tools to understand characteristics, structure, and behavior of code. These include tools such as target system debuggers, code navigators (e.g., Understand), or flowcharters.
IV&V tool effectiveness	An assessment of effectiveness of all tools employed to support code validation including analytical tools, formal method support, automated static analyzers, requirements trace etc.
Analyses employed	An assessment of the effectiveness of the types of analyses planned such as Lint analyses, code reading, formal method, etc.
Developer review participation	Assessment of effectiveness of plans for participating in developer milestone reviews, inspections, walkthroughs etc. Assessment would include timing in which IV&V enters the project (SRR, SDR SSR, etc) and its impact on effectiveness.
IV&V techniques employed effectiveness	Overall level and effectiveness of defect removal techniques used by IV&V

B.4 Test Issue Subnet

Name	Characteristic Description
Test Defect Introduction	
Test Plan Quality	Quantitative estimate of test plan quality. Consider whether the test plan contains clear definition of test cases, their objectives, complete description of test equipment and its capabilities, coverage of requirements, and a schedule for test.
Requirements Stability	Estimate of rate of change of the requirements. If the requirements are changing rapidly, it becomes difficult to define test cases to verify the requirements. The requirements stability comes from the requirements phase BBN.
Test Description Quality	Estimate of the quality of the test description. Does this document contain a good design for the test cases? Does it contain details as to how each test case will be implemented, what equipment it will use and how, what constraints are imposed, success criteria, inputs for each test case, models to be used and their required fidelity, data to be collected and how that will be accomplished (method for access of data), required control of target computer, and analyses to be completed and how that will be accomplished.
Test Procedures Stability	How stable are the procedures? This is driven by the test plan quality, test description quality, requirements stability, and the proneness of the test group to make errors. The more stable, the fewer changes and the lower the risk of introduction of new errors.
Problem Complexity	Qualitative estimate of overall system/problem complexity. Related to technical difficulty to define, required interfaces (developer and system), and unity of users. Not correlated with code complexity metrics. Simple (10) to complex (1)
Test Problem Space	Overall susceptibility of the problem space to introduction of test defects. Represents the difficulty, based on the complexity of the problem being solved, and the quality and stability of documentation describing the problem, in deriving a correct implementation (set of test procedures).
Dev staff experience level	Average experience level of development staff, not specific to the problem domain, but overall experience in software testing for the domain type (e.g., real-time embedded flight, financial, ground, manned, etc). Includes staff experience with the language chosen for implementation and the operating system in use. Also includes experience with the processor to be used, and the hardware to be interfaced with. Also includes experience with the test equipment to be used and development of test cases for the type of software being tested.
Dev domain experience	Average experience of development staff in testing the specific application domain (e.g., laser guidance system, space telescope, crew rescue, etc). Consider all individual domains within the system (e.g., space telescope will require GN&C, optics, propulsion, system management, telemetry, etc).
Dev schedule pressure	How much margin is in the testing schedule? Assessment of flexibility in end date. A higher number indicates developers have plenty of time to complete their work, a lower number

	indicates developers are consistently rushed to deliver products.
Dev budget margin	How tight is the development budget? Assessment of flexibility in cost growth.
External Constraint Pressure	Overall influence level of external constraints related to schedule and budget. How strong is the pressure to proceed without fix in spite of schedule or budget problems?
Process effectiveness actions	How much emphasis does development management place on quality (as opposed to productivity)? For example, does development management ensure that review action items are tracked and encourage extra analysis of suspected problems? Assessment of effectiveness of process problem reports (e.g., a formal mechanism to document, correct and publish discrepancies in following the process), process improvement actions, activity of a board to assess effectiveness of process, etc. An assessment of the 'aliveness' of the developer process and attention to making it work to produce better products.
Dev quality organization	How effective is the embedded quality organization? This measure includes consideration of size, breadth and depth of capability applied to this project, and level of authority granted to the quality organization applied to this project.
Process Adherence	How well is the development staff likely to adhere to the documented test development process? Should be based on knowledge of schedule and budget constraints, level of activity related to process effectiveness, and the quality of organization enforcing the process.
Turnover	Experienced or historical rate of change of staff involved with test development. A higher number indicates little turnover, a lower number indicates a lot of turnover.
Staff level	Assessment of whether the quantity and distribution across domains of staff is sufficient for the problem space. An adequate staff level should receive a very high score (not average). The worst staff level with respect to work required ever seen by the evaluator would receive a score of 1.
Resource Availability	Measure of the degree to which the size of the development staff is sufficient and stable in terms of longevity on the project.
Staff Ability	Overall ability level of the test development staff, in terms of size, development experience, domain experience and turnover with respect to the problem at hand.
Process definition, product standards, quality criteria	This is an overall assessment of the effectiveness of the development process related to development of test. The assessment should include methods for test procedure development and debug, effective use of the test equipment, effective methods for data acquisition, test execution, documentation, and methods for development of analyses for verification. It should correlate fairly well with CMM level, but includes an assessment of what is really happening in addition to what is documented.
Dev tools	Degree to which the developer uses tools in developing and analyzing testing. Tools here include requirements coverage support, traceability tools, code coverage support, test case generation, execution control, etc.
Process Rigor	How rigorous can the process be expected to be? Is the process well founded (related to CMM level), is it supported by a good set of tools, does the development organization pay

	attention to and follow the process?
Test quality	Overall relative measure of test quality, and hence, probable relative defect density.

Developer Test Defect Removal Efficiency	
External Constraint Pressure	Similar to the cost and schedule pressure on test procedure development. In some cases, the developer test procedure/checking process is more tightly constrained than test procedure development because of late development.
Process effectiveness actions	Same as process effectiveness actions input for test procedure development unless there is something unique about the group that reviews the products to find defects.
Defect Removal Effort/Focus	How much effort or focus does the developer apply to defect identification and removal during the test phase? Are external organizations involved to make sure test procedures are right, is there a rush to complete deliverables, is the defect removal process effective and adhered to?
Simulation Validity	Degree of verification of simulations used to test flight software requirements. The flight software is not verified unless the models used to verify it are shown to be correct.
Test Case Automation	Degree and effectiveness of use of automated test case generation and requirements coverage analysis tools.
Analytical tool use	Degree to which the development organization uses analytical tools to understand and validate design and implementation. These include tools such as target system control for breakpoints and data access or code navigators. This also includes tools to support understanding of interfaces (in particular sequence and timing of data exchange, interrupt timing and frequency, potential range and quantity of data, operational dynamics of interfacing hardware, and environmental requirements of interfacing hardware) to support the definition of effective test cases. Coverage of novel or complex areas.
Tools effectiveness	Overall level and effectiveness of tool use in the test procedure analysis process for removal of defects.
Review quality	Quality of the embedded review process. For example, do the developers do formal, detailed test procedure reviews? Do they use entry and exit criteria, track and board issues, etc? Do they hold walkthroughs with broad scope support?
Techniques Employed effectiveness	Overall level and effectiveness of defect removal techniques used by the developer.

IV&V Test Defect Removal Efficiency	
Data timing	Relative measure of timeliness of data delivery by the developer to IV&V. Late data delivery places time restrictions on IV&V and can impede IV&V.
Data completeness	Measure of completeness of data submitted to IV&V. For example, if important information is withheld due to proprietary concerns, that will impede IV&V. Another example is that test procedures are delivered with sections incomplete or many TBDs or test description does not have content expected for the point in lifecycle at which it was delivered.
Data availability	Likelihood that sufficient and timely input data will be available to IV&V when it is needed.
IV&V environment	Overall operating environment given to IV&V. Are artifacts delivered on time and do they contain what is needed for efficient evaluation, are issues considered fairly and in a timely manner?
Direct dev access	How much access does IV&V have to the developers? If IV&V has to work through several bureaucratic levels to get information from the developers or discuss issues or risks, that will decrease IV&V effectiveness.
Project acceptance	How well does the project accept the IV&V participation? A high score indicates the project participants exhibit a belief that use of IV&V will lead to higher mission success probability. Project management has issued directives exhibiting the right intent in dealing with the IV&V participation.
Dev cooperation	How cooperative are developers, in general, in responding to IV&V requests and suggestions? Do all IV&V issues get immediate attention, or does the development organization tend to ignore or avoid dealing with IV&V concerns? Developers exhibit attention to IV&V concerns and timely/meaningful response. While project acceptance is having the right intent, cooperation is doing the right thing.
Project/IV&V interface	How efficient is the interface between the developer and IV&V considering access, cooperation and acceptance.
IV&V experience level	Average experience level of IV&V staff. This is a measure of how much experience the IV&V staff has in the area of IV&V and testing activities. This does not consider domain experience level which is considered in another input. This considers experience in implementation language, operating system in use, processing platform and hardware interfaces.
IV&V domain experience/expertise level	Degree of experience of IV&V staff with the application domain. Consider all individual domains within the system (e.g., GN&C, power, C&DH, ECLSS, terrain mobility, thermal, etc). This item should evaluate the extent of applicable domain knowledge within the IV&V staff.
IV&V Staffing level	How appropriate is the size of the IV&V staff to the analysis tasks that need to be performed based on the CARA results? Too few (or too many) would lower the rating. An adequate staff level should receive a very high score (not average). Descoping the tasks from the CARA results to match a low staff level would get a low score.
Schedule pressure	How much schedule pressure does IV&V face. This factor could be correlated with data timing, but not necessarily. A low score

	indicates there was heavy schedule pressure. A high score indicates there was little schedule pressure.
Resource availability	Availability of all human resources needed to perform IV&V. A sufficient set of personnel is available to perform the IV&V activities including consideration of the schedule pressure from the project (e.g., very short turnaround of document review expected), size of staff and personnel turnover.
IV&V Staff ability	Overall ability of IV&V staff to perform the IV&V tasks defined by the CARA analysis.
Test Automation	Degree and effectiveness of use of coverage analysis tools.
Understanding of developer tool use	Degree to which the IV&V organization understands the tools to be used by the developer and is able to assess effectiveness of usage.
Tool Use	An assessment of effectiveness of all tools employed to support test procedure analysis including automation tools and understanding of developer tools.
Analyses employed	An assessment of the effectiveness of the types of analyses planned such as scenario analyses, simulation, or similarity to determine completeness and correctness of test procedures.
Developer review participation	Assessment of effectiveness of plans for participating in developer milestone reviews, inspections, walkthroughs etc. Assessment would include timing in which IV&V enters the project (SRR, SDR SSR, etc) and its impact on effectiveness.
IV&V techniques employed effectiveness	Overall level and effectiveness of defect removal techniques used by IV&V

B.5 Integration Issue Subnet

Parameter	Characteristic Name and Description
	Integration Defect Introduction
System Requirements Stability	Quantitative estimate of the quality and stability of the system requirements being verified in the integration testing.
Stakeholder Involvement	Degree to which the system users are involved in the integration test process and the timeliness of that involvement. Note that a high score here requires involvement or representation (by users or system engineering, for example, with equivalent knowledge) of all key system users, not just system operators.
Integration Stability	Probability of change to the driving requirements and user desires affecting the development of the integration test procedures.
Integration Test Plan Quality	Quantitative estimate of integration test plan quality. This should include completeness of test case definition in terms of needed inputs, success criteria, needed equipment, needed data access, and requirements coverage.
Integration Test Procedures Stability	How stable are the procedures? This is driven by the integration test plan quality, integration test description quality, system and interface requirements stability, and the proneness of the test group to make errors. The more stable, the fewer changes and the lower the risk of introduction of new errors.
Problem Complexity	Qualitative estimate of overall system/problem complexity. Related to technical difficulty to define, required interfaces (developer and system), and unity of users. Not correlated with code complexity metrics. Simple (10) to complex (1)
Integration Test Equipment Attributes	Rating of capability and complexity of equipment and simulations to be used in performing system level verification. Capability is rated in terms of ability to perform all functions needed as a part of requirements verification. Complexity is rated in terms of ease and reliability of use of the capabilities.
Integration Test Problem Space	Overall susceptibility of the problem space to introduction of test defects. Represents the difficulty, based on the complexity of the problem being solved, and the quality and stability of documentation describing the problem, in deriving a correct implementation (set of test procedures).
Dev staff experience level	Average experience level of development staff, not specific to the problem domain, but overall experience in software testing for the domain type (e.g., real-time embedded flight, financial, ground, manned, etc). Includes staff experience with the language chosen for implementation and the operating system in use. Also includes experience with the processor to be used, and the hardware to be interfaced with. Also includes experience with the test equipment to be used and development of test cases for the type of software being tested.
Dev domain experience	Average experience of development staff in testing the specific application domain (e.g., laser guidance system, space telescope, crew rescue, etc). Consider all individual domains within the system (e.g., space telescope will require GN&C, optics, propulsion, system management, telemetry, etc).
Dev schedule pressure	How much margin is in the integration testing schedule?

	Assessment of flexibility in end date. A higher number indicates developers have plenty of time to complete their work, a lower number indicates developers are consistently rushed to deliver products.
Dev budget margin	How tight is the development budget? Assessment of flexibility in cost growth.
External Constraint Pressure	Overall influence level of external constraints related to schedule and budget. How strong is the pressure to proceed without fix in spite of schedule or budget problems?
Process effectiveness actions	How much emphasis does development management place on quality (as opposed to productivity)? For example, does development management ensure that review action items are tracked and encourage extra analysis of suspected problems? Assessment of effectiveness of process problem reports (e.g., a formal mechanism to document, correct and publish discrepancies in following the process), process improvement actions, activity of a board to assess effectiveness of process, etc. An assessment of the 'aliveness' of the developer process and attention to making it work to produce better products.
Dev quality organization	How effective is the embedded quality organization? This measure includes consideration of size, breadth and depth of capability applied to this project, and level of authority granted to the quality organization applied to this project.
Process Adherence	How well is the development staff likely to adhere to the documented integration test development process? Should be based on knowledge of schedule and budget constraints, level of activity related to process effectiveness, and the quality of the organization enforcing the process.
Turnover	Experienced or historical rate of change of staff involved with integration test development. A higher number indicates little turnover, a lower number indicates a lot of turnover.
Staff level	Assessment of whether the quantity and distribution across domains of staff is sufficient for the problem space. An adequate staff level should receive a very high score (not average). The worst staff level with respect to work required ever seen by the evaluator would receive a score of 1.
Resource Availability	Measure of the degree to which the size of the integration test development staff is sufficient and stable in terms of longevity on the project.
Staff Ability	Overall ability level of the integration test development staff, in terms of size, development experience, domain experience and turnover with respect to the problem at hand.
Process definition, product standards, quality criteria	This is an overall assessment of the effectiveness of the development process related to development of integration test. The assessment should include methods for integration test procedure development and debug, effective use of the integration test equipment, effective methods for data acquisition, test execution, documentation, and methods for development of analyses for verification. It should correlate fairly well with CMM level, but includes an assessment of what is really happening in addition to what is documented.
Dev tools	Degree to which the developer uses tools in developing and analyzing integration testing. Tools here include requirements coverage support, traceability tools, interface coverage support,

	test case generation, execution control, etc.
Process Rigor	How rigorous can the process be expected to be? Is the process well founded (related to CMM level), is it supported by a good set of tools, does the development organization pay attention to and follow the process?
Integration Test quality	Overall relative measure of integration test quality, and hence, probable relative defect density.

Developer Integration Defect Removal Efficiency	
External Constraint Pressure	Similar to the cost and schedule pressure on integration test procedure development. In some cases, the developer test procedure/checking process is more tightly constrained than test procedure development because of late development.
Process effectiveness actions	Same as process effectiveness actions input for integration test procedure development unless there is something unique about the group that reviews the products to find defects.
Defect Removal Effort/Focus	How much effort or focus does the developer apply to defect identification and removal during the integration test phase? Are external organizations involved to make sure integration test procedures are right, is there a rush to complete deliverables, is the defect removal process effective and adhered to?
Simulation Validity	Degree of verification of simulations used to test system requirements. The models used to verify requirements must, themselves, be verified.
Test Case Automation	Degree and effectiveness of use of automated test case generation and requirements coverage analysis tools.
Analytical tool use	Degree to which the development organization uses analytical tools to understand and verify design and implementation. These include tools such as target system control for breakpoints and data access or code navigators. This also includes tools to support understanding of interfaces (in particular sequence and timing of data exchange, interrupt timing and frequency, potential range and quantity of data, operational dynamics of interfacing hardware, and environmental requirements of interfacing hardware) to support the definition of effective test cases. Coverage of novel or complex areas.
Tools effectiveness	Overall level and effectiveness of tool use in the integration test procedure analysis process for removal of defects.
Review quality	Quality of the embedded review process. For example, do the developers do formal, detailed code reviews? Do they use entry and exit criteria, track and board issues, etc? Do they hold walkthroughs with broad scope support?
Techniques Employed effectiveness	Overall level and effectiveness of defect removal techniques used by the developer.

IV&V Integration Defect Removal Efficiency	
Data timing	Relative measure of timeliness of data delivery by the developer to IV&V. Late data delivery places time restrictions on IV&V and can impede IV&V.
Data completeness	Measure of completeness of data submitted to IV&V. For example, if important information is withheld due to proprietary concerns, that will impede IV&V. Another example is that integration test procedures are delivered with sections incomplete or many TBDs or test description does not have content expected for the point in lifecycle at which it was delivered.
Data availability	Likelihood that sufficient and timely input data will be available to IV&V when it is needed.
IV&V environment	Overall operating environment given to IV&V. Are artifacts delivered on time and do they contain what is needed for efficient evaluation, are issues considered fairly and in a timely manner?
Direct dev access	How much access does IV&V have to the developers? If IV&V has to work through several bureaucratic levels to get information from the developers or discuss issues or risks, that will decrease IV&V effectiveness.
Project acceptance	How well does the project accept the IV&V participation? A high score indicates the project participants exhibit a belief that use of IV&V will lead to higher mission success probability. Project management has issued directives exhibiting the right intent in dealing with the IV&V participation.
Dev cooperation	How cooperative are developers, in general, in responding to IV&V requests and suggestions? Do all IV&V issues get immediate attention, or does the development organization tend to ignore or avoid dealing with IV&V concerns? Developers exhibit attention to IV&V concerns and timely/meaningful response. While project acceptance is having the right intent, cooperation is doing the right thing.
Project/IV&V interface	How efficient is the interface between the developer and IV&V considering access, cooperation and acceptance.
IV&V experience level	Average experience level of IV&V staff. This is a measure of how much experience the IV&V staff has in the area of IV&V and integration testing activities. This does not consider domain experience level which is considered in another input. This considers experience in implementation language, operating system in use, processing platform and hardware interfaces.
IV&V domain experience/expertise level	Degree of experience of IV&V staff with the application domain. Consider all individual domains within the system (e.g., GN&C, power, C&DH, ECLSS, terrain mobility, thermal, etc). This item should evaluate the extent of applicable domain knowledge within the IV&V staff.
IV&V Staffing level	How appropriate is the size of the IV&V staff to the analysis tasks that need to be performed based on the CARA results? Too few (or too many) would lower the rating. An adequate staff level should receive a very high score (not average). Descoping the tasks from the CARA results to match a low staff level would get a low score.
Schedule pressure	How much schedule pressure does IV&V face. This factor could

	be correlated with data timing, but not necessarily. A low score indicates there was heavy schedule pressure. A high score indicates there was little schedule pressure.
Resource availability	Availability of all human resources needed to perform IV&V. A sufficient set of personnel is available to perform the IV&V activities including consideration of the schedule pressure from the project (e.g., very short turnaround of document review expected), size of staff and personnel turnover.
IV&V Staff ability	Overall ability of IV&V staff to perform the IV&V tasks defined by the CARA analysis.
Test Automation	Degree and effectiveness of use of coverage analysis tools.
Understanding of developer tool use	Degree to which the IV&V organization understands the tools to be used by the developer and is able to assess effectiveness of usage.
Tool Use	An assessment of effectiveness of all tools employed to support integration test procedure analysis including automation tools and understanding of developer tools.
Analyses employed	An assessment of the effectiveness of the types of analyses planned such as scenario analyses, simulation, or similarity to determine completeness and correctness of test procedures.
Developer review participation	Assessment of effectiveness of plans for participating in developer milestone reviews, inspections, walkthroughs etc. Assessment would include timing in which IV&V enters the project (SRR, SDR SSR, etc) and its impact on effectiveness.
IV&V techniques employed effectiveness	Overall level and effectiveness of defect removal techniques used by IV&V