# MANAGING ITERATIONS AND REWORK IN INCREMENTAL PRODUCT DEVELOPMENT

K. Aravindhan, Dr. Trishit Bandyopadhyay, Dr. Supriya Kumar De, Dr. Mahesh Mehendale,

***Abstract*— In order to meet the market requirements in terms of cost and time most of the organizations have adopted the modular product architecture in which the final product is made up of several smaller components – either reused or new. In an incremental product development a few new components are incorporated to existing components (from previously developed product) to enhance the product features in a cost effective manner. To further enhance the time to market, organizations develop the components concurrently. In this scenario typically two types of defects are seen – defects in individual components and defects at a product level due to interaction among the components. The managerial challenge is to provide the high degree of uniqueness that seems necessary for competitive success while retaining the scale economies required for low cost. The aim of this research is to study the rework in the context of incremental product development where certain components are new and certain components are reused. The impact on rework effort arising from new, reused components and complexity of the product is studied. The study is carried out in the context of System on a Chip (SoC) design and development at Texas Instruments.**

***Keywords*—Iteration, Product Design, Rework, Reuse, Incremental Product development**

## INTRODUCTION TO INCREMENTAL PRODUCT DEVELOPMENT

Designing new products is usually done through changes to existing products (Otto and Wood 2001). Nichols (1990), for example, reports that 80% of all parts of American cars in the 1980s were carried over from previous designs (40– 50% for similar Japanese cars). Reuse helps us to bring products to market faster and also helps to keep the quality of new products under control. Sometimes one may not have enough time to do a full reliability testing and rely on the fact that most of the pieces in the product are reused from prior stable products. Reuse affords economy of scale in production and reduced design time compared to new development. In many markets the customers also do not want a completely changed product as it may have robustness and reliability issues. Depending on their newness to the company and marketplace, product innovations can be incremental or radical (Henderson and Clark, 1990; McDermott, 1999; Hauser et al., 2006). Radical innovation often requires developing products with an entirely new set of performance features (Leifer et al., 2000; Zhou et al., 2005). On the other hand, an extension or improvement of existing products is termed as incremental product innovation. Incremental product innovation plays a major role in the success of many organizations since the majority of so called ‘new’ products are in fact reworked versions of existing products (Ali, 1994; Griffin, 1997; Grupp and Maital, 2001). Understanding the impact of the changes in an incremental product development seems to be necessary given the fact that most of the companies adopt the incremental product development as the major strategy.

## COMPONENT BASED DEVELOPMENT FLOW

The notion of modularization as a strategy emerged during the 1960s, and many optimization models were introduced to investigate the modularity problem (Evans, 1963; Passy, 1970; Shaftel, 1971) and the modular production concept (Starr, 1965), which describes the essence of how to design, develop, and produce parts that can be combined in a maximum number of ways to deal with consumers' demand for variety and uniqueness. The constituent components, which may be standard (STD) or new to the firm (NTF), and how they are linked to one another determine the performance and cost benefits of present and future generations of product architectures (Mikkola et.al 2003). More recently, higher product complexity, increased competition, customer expectations for customization, shortened reaction times, and larger numbers of activities and amounts of information to coordinate have increased the need for a systematic approach to managing product development. Concurrent Engineering and Integrated Product and Process Development have increased the overlap among PD activities, dramatically increasing the coordination challenge (Browning et al. 2002). The studies of the world automobile industry, companies with short development lead times overlapped their development activities with frequent information transfer. Clark and Fujimoto 1991 call this combination of activity overlap and intensive communication "integrated problem solving." In order to accelerate the time to market, firms attempt to overlap the different activities in product design and development – leading to iterative overlapped development. Understanding the impact of the changes in an incremental product development seems to be necessary given the fact that most of the companies adopt the incremental product development as the major strategy. Managers hence need to decide on the number of standard and new to the firm components needed to be designed to meet the product requirements.

## STUDY OF SoC PROJECTS AT TEXAS INSTRUMENTS

The aim of this research is to study the rework in the context of incremental product development where certain components are new and certain components are reused. The impact on rework effort arising from new, reused components and complexity of the product is studied. The study is carried out in the context of System on a Chip (SoC) design and development at Texas Instruments.

### *1)STEP BY STEP ANALYSIS OF THE 10 CASES*

The first step is to study the development flow from the project plan, the process methodology and process compliance expected to be followed in the product development flow. Sources of information – project kick off meeting where process compliance expected in the product development is documented. This does not differ from project to project.

### *B. Step 1 – Identify the new components and Reused components in the design*

Study the project commissioning document to find the number of components that were to be taken from previous designs and new components to be developed for the specific SoC. The reuse is reported in the project commissioning document and also is an important factor in the overall cost of the design. The information about reused components and new components is also found in the program reviews of the SoC.

### *C. Step 2 – Identify the count of defects – IP defects and system defects*

The next step involved is in counting the number of component defects and System Defects. These were obtained from the defect data base of Texas Instruments which is well maintained. Also comparison of the defect data with the monthly program reviews were done to ensure all the defects are accounted for. Component defects were identified by seeing whether the defects were filed against the components teams or they were filed against the Silicon (SoC) team. The defects filed against component teams were to be fixed by re-releasing the components while the defects filed against the Silicon team were to be fixed by the Top level design team. The count of defects were periodically reported in the monthly reviews held with senior management and we ensured that the data in the defect data base matched with the monthly review reports. This is again a standard process that is uniform across the SoC design projects in Texas Instruments.

### *D. Step 3 – Build relationship using regression technique*

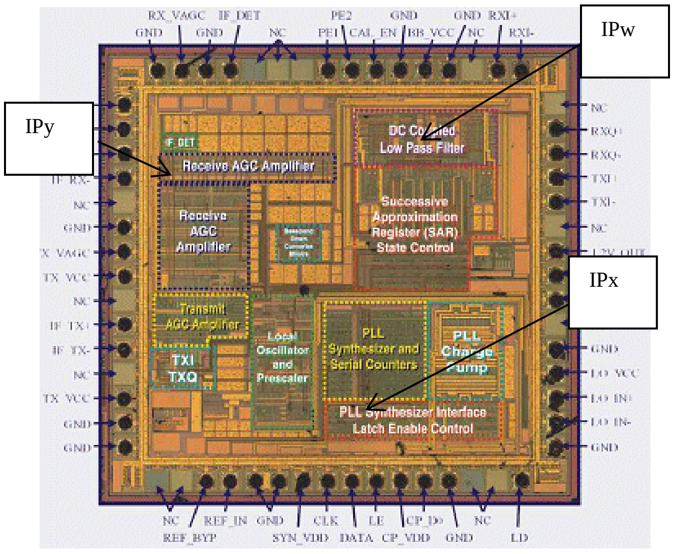
Build relationship between defects and number of new and reused components in the product and product complexity. Given that the defect data is a count variable generalized linear modelling is used and to handle overdispersion the quasipoisson family is used for generalized linear modelling.

### *E. Step 4 – Validate the results through boot strapping technique*

Bootstrapping is general approach to statistical inference based on building a sampling distribution for a statistic by resampling from the data at hand (John Fox 2002). The term ‘bootstrapping,’ due to Efron (1979), is an allusion to the expression ‘pulling oneself up by one’s bootstraps’ – in this case, using the sample data as a population from which repeated samples are drawn. 2 types of Boot strapping used in the analysis of the cases in this paper are Randox-x Resampling and Fixed-x Resampling. One can treat the predictors as random, potentially changing from sample to sample. Random-x resampling is also called case resampling, and fixed-x resampling is also called model-based resampling.

## CHIP LAYOUT WITH IPs

A system on chip (SoC) consists of many components which are referred to as IPs. The system is redesigned for a new end product by adding new components (components henceforth will be called as IP – a terminology used in SoC design world for components) and reusing components from existing designs. The figure 1 shows a chip layout with many IPs (Receive AGC Amplifier, DC Coupled Low Pass Filter, PLL Synthesizer etc – reference http://www.design-reuse.com/articles/19947/ip-core-protection-identification.html).



The SoC design is made up of IP design and Top Level design. The Top level is the interconnections between IPs to get the IPs connected with each other to provide the needed functionality. One can consider the IPs as tyres, doors, seats etc. in a car and the Top level as the entire car fitted with all the needed components.

## IP DEFECTS

The defects analyzed in this paper are of 2 types. IP defects are the defects that were found at the final system level verification after they have been verified by the IP team. We define these as defects leaked through the IP team and found by the system level verification when all the IPs are integrated together and found during integration testing.

SYSTEM DEFECTS

The system defects are defects found in the top level SoC when the IPs are integrated together. These defects are not fixed in the IPs but in the top level design. The top level defects (system defects) are primarily defects due to, error in interconnections between IPs or logic errors.

## ANALYSIS OF DATA

The system defects that are counted in the cases refer to top level defects. These defects are not attributed to the IPs and are hence fixed by the top level design team. The defects can lead to significant rework and redesign. IP usage familiarity helps in reducing these defects since the reused IPs have been used in a chip previously and interfaces have been correctly verified in prior SoCs.

*Table 1 : Data from 10 SoC products from Texas Instruments*

| TOTAL | NIPs | RIPs | PRNR | SIZE | IPD |
|---|---|---|---|---|---|
| 84 | 1 | 75 | 75 | 78 | **1** |
| 96 | 14 | 63 | 882 | 62 | **5** |
| 457 | 49 | 60 | 2940 | 113 | **32** |
| 375 | 26 | 55 | 1430 | 49 | **8** |
| 62 | 4 | 37 | 148 | 75 | **4** |
| 220 | 10 | 34 | 340 | 44 | **23** |
| 675 | 25 | 62 | 1550 | 78 | **399** |
| 450 | 36 | 40 | 1440 | 104 | **79** |
| 755 | 79 | 0 | 0 | 70 | **341** |
| 241 | 15 | 80 | 1200 | 95 | **2** |

RIPs – Number of Reused IPs
NIPs – Number of New IPs
SIZE – Size of the SoC in sq.mm
IPD – IP Defects
TOTAL – System Defects – SD
PRNR – Product of Number of New IPs and Reused IPs

## SUMMARY OF ALL THE MODELS

**MODEL 1**

$$\log(TD) = 0.085082 \times NIPs + 0.070661 \times RIPs - 0.0005878 \times PRNR$$

| | **Estimate** | **Std.Error** | **t-value** | **Pr(>\|t\|)** | |
|---|---|---|---|---|---|
| **NIPs** | **0.085082** | **0.010094** | **8.429** | **6.52E-05** | *** |
| **RIPs** | **0.070666** | **0.013301** | **5.313** | **0.00111** | ** |
| **PRNR** | **-0.00059** | **0.000473** | **-1.244** | **0.25351** | |

*Table 2 : GLM results of model 1*

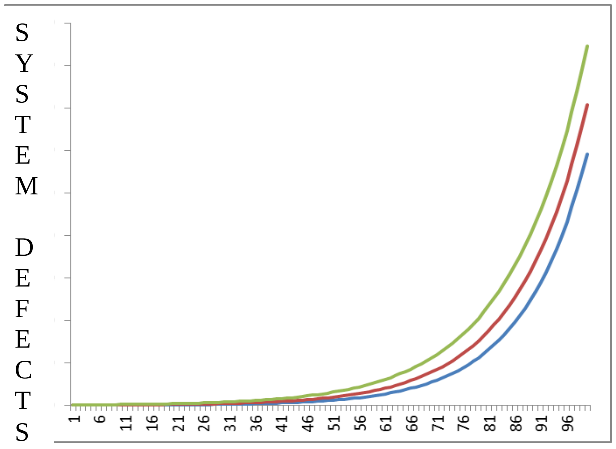
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS (Based on 2000 bootstrap replicates)

| Coefficients | Level | Percentile range from boot strapping fixed X | Percentile range from boot strapping random X | Value from the 10 cases |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Coeff 1 | 95% | (0.0849, 0.0855 ) | (0.0838, 0.4709) | 0.085082 |
| Coeff 2 | 95% | (0.0700, 0.0720) | (0.0400, 0.1120) | 0.070661 |
| Coeff 3 | 95% | (-0.0006, -0.0006 ) | (-0.0071, 0.0011) | -0.0005878 |

- The relationship is exponential – the system defects increase exponentially with the increase in number of new IPs, reused IPs. The contribution of new IPs to the system defects as compared to reused IPs is seen to be more (0.0850821 vs. 0.070661).
- The coefficient values fall within the 95% percentile range for both fixed-X and random-X bootstrapping values.
- 8.5 is the percentage change in System defects due to adding 1 new IP when the number of reused IP is at zero
- 7 is the percentage change in System defects due to adding 1 reused IP when the number of New IPs is at zero
- -0.00058 is the amount the slope of New IPs on System defects changes when Reused IP increases by one unit
- (0.085 – 0.00058R) * 100 : the impact of one unit increase of New IPs will have on System Defects - moderated by number of reused IPs
- (0.070 – 0.00058N) * 100 : the impact of one unit increase of Reused IPs will have on System Defects - moderated by number of new IPs

*Fig 2 : PLOT OF SYSTEM DEFECTs vs NEW IPs (For 3 different values of RIPs)*



Managerial insights: The number of new or Reused IPS increases the system defects exponentially. Clearly the complexity of design increases exponentially with the increase in number of components (new + reused IPs). This is in line with the conclusions made in literature as well. The incremental product design needs to be carefully decided as the number of new IPs can cause more increase in defects (vs. reused IPs) and thereby increasing the overall development time. The reused IPs contributes to lesser extent as compared to the new IPs.

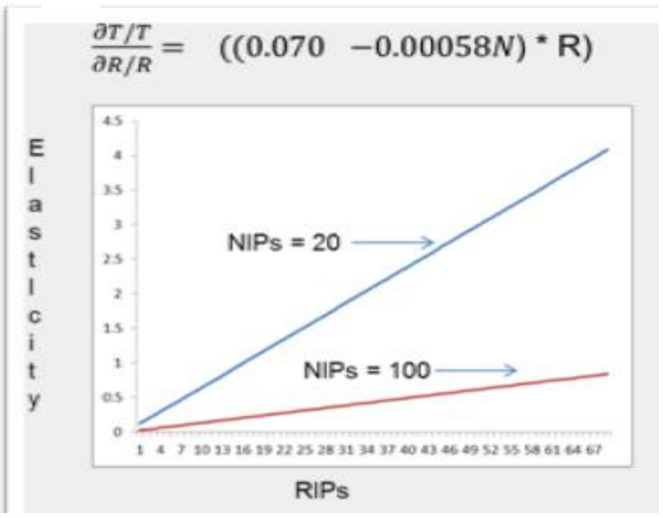


*Fig 3 : SYSTEM DEFECTs ELASTICITY WITH REUSED IPs*

Managerial insights – The rate of change of system defects w.r.t rate of change of reused IPs (defect Elasticity) is moderated by the number of new IPs. If the new IPs is higher the rate of change of system defects w.r.t rate of change of reused IPs is lower than if the number of new IPs are lower (red line vs blue line in the plot on the left hand side). The reasoning is that the percentage change in the Total IPs is lesser with higher New IPs with increase in reused IPs. The percentage change in the Total IPs is higher with the lower New IPs with increase in reused IPs.

## MODEL 2- IP DEFECTS AS A FUNCTION OF NEWIPS, REUSEIPS

The analysis below carried out to study the relationship between IP Defects using the Number of new IPs and reused IPs.

MODEL 2

$$\log(IPD) = 0.07159 \times NIPs + 0.0353294 \times RIPs$$

$$IPD = e^{0.07159\text{NIPs}} \times e^{0.03532942\text{RIPs}}$$

| | Estimate | Std.Error | t-value | Pr(>\|t\|) | |
|---|---|---|---|---|---|
| NIPs | 0.0716 | 0.01294 | 5.532 | 0.000553 | *** |
| RIPs | 0.03533 | 0.01497 | 2.36 | 0.045991 | * |

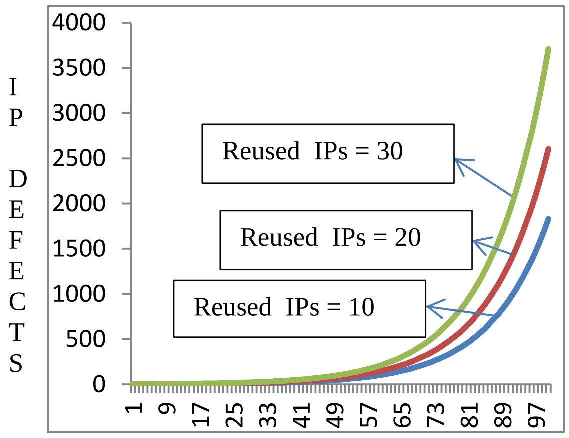
*Table 3 : GLM results of model 2*

RANDOM X – BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS (Based on 2000 bootstrap replicates)

| Coefficients | Level | Percentile range from boot strapping random X | Values from 10 cases |
|---|---|---|---|
| Coefficient 1 | 95% | (0.0212, 0.1292) | 0.07159 |
| Coefficient 2 | 95% | (-0.0095, 0.0668) | 0.0353294 |

Clearly from the boot strapping results (Random X) the coefficients fall within percentile limits and hence model 2 can be considered as a reasonably good model for IP defects.

- The contribution of Number of new IPs (NIPs) is 0.07159. The contribution of Number of reused IPs (RIPs) is 0.0353294. This is clearly significantly lesser than the contribution to the system defects in model 1.
- The ratio of the impact of NewIPs as compared to reused IPs is a factor of 2 while in model 1 the impact of reused IPs is very close to the new IPs.

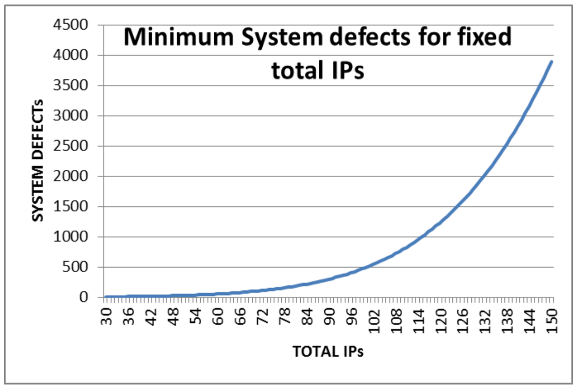


- Clearly from the boot strapping results (Random X) the coefficients fall within percentile limits and hence model 2 can be considered as a reasonably good model for IP defects.

*Fig 4 : MODEL 2 – IP Defects vs. Number of New IPs for 3 different values of Reused IPs*

Managerial Insights – it is very clear that the New IPs contribute to IP defects to a greater extent than the reused IPs. When compared to the system defects (model 1) the impact of reused IPs is much lesser to the IP defects. One can say that the IP defects are more impacted by number of new IPs as compared to reused IPs but when it comes to system defects reused IPs impact almost equally as the new IPs. The incremental product design needs to be carefully decided as the number of new IPs can cause exponential increase in IP defects and thereby increasing the overall development time. Model 2 provides insight into the impact of combination of new and reused IPs.

*Fig 5 : Model 1 - RIPs + NIPs = Constant and finding minimum system defects*



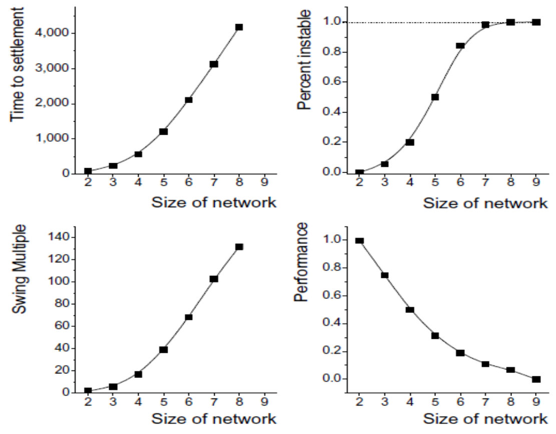
The system defects increases exponentially with the increase in Total IPs. This clearly implies that the total development time will also increase exponentially due to the effort needed for fixing the defects uncovered. Adding more IPs in fact results in escalating increase in cost due to the defects found during the development time. Additionally one can identify the optimum number of Total IPs that a particular product should have in-order to control the total effort. Hence the managerial decision is very important to limit the total IPs.

## CONCLUSION

### Conclusion 1 - Exponential growth of defects with increase in Total Components

The number of components and parts in a product has been generally used to describe the complexity of the product in terms of its size; the more components to consider in a product, the greater the complexity in product design, production, and supply chain. Results clearly indicate that there is a relationship between defects and Total IPs. The system on chip products are very similar and one can conclude that the number of IPs in the SoC drives productivity. As product complexity increases, the life cycle cost of the product will increase; a complex product typically results in complicated and costly product design and development processes, causing inefficiencies in the product realization phase (Nihal Orfi et.al 2011). Empirical studies show that there is a strong positive correlation between the measured complexity and the number of errors or the productivity drop of the manufacturing system (Martin and Ishii 1996, Sarkis 1997, Shibata et al. 2003, Kinnunen 2006). One of the key findings is that the increase in complexity is exponential in relationship. Real world study of defects in SoCs have shown additional proofs for these. Complex systems are characterized as "made up of a large number of parts that interact in nonsimple ways _ _ _ [such that] given the properties of the parts and the laws of their interactions, it is not a trivial matter to infer the properties of the whole" (Simon 1969, p. 195). In particular, the performance properties of a complex system represent a "rugged landscape": Interactions among a multitude of decision variables weaken the correlation between the performance values of neighboring design choices. Thus, the highest performance peaks cannot be identified or found with local (incremental) search (Kauffman 1993, Levinthal 1997). The

*Fig 6 : Base case (adopted from Jurgen et. al 2003)*



above described difficulty in designing such complex systems manifests itself in widespread performance problems—budget and schedule overruns, missed specifications (e.g., Morris and Hugh 1987, Terwiesch and Loch 1999, Tatikonda and Rosenthal 2000), and management frustration with "performance oscillations." Jurgen Mihm et.al 2003 results offer three important insights. First, they show how easily a rugged performance landscape arises, even from simple components with single-peaked performance functions, if the components are interdependent. The system becomes highly nonlinear even if the interdependencies are (piecewise) linear. Second, they characterize the dynamic behavior of the system, arising from the designers making successive local component decisions over time, taking into account the current status of the surrounding components. The time for the system to settle at the design fixed point grows as a function of the network size *N*. The upper right panel shows the percentage of networks that become "unstable." This fraction grows with network size and soon approaches 1. The reason for this lies, again, in the system feedback: Design decisions move in cycles as interdependent components keep changing.

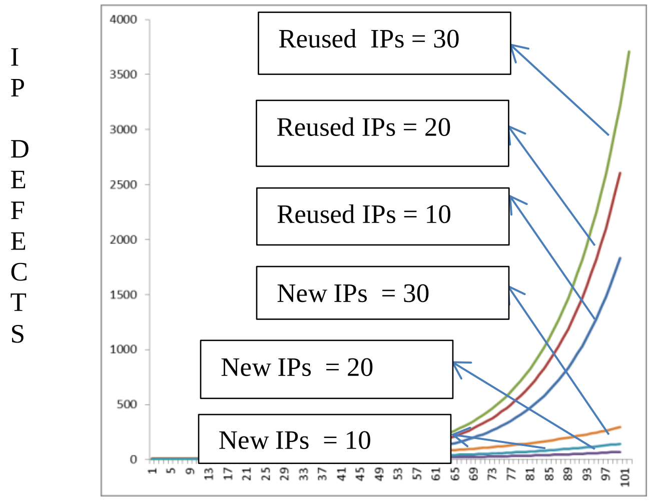
<u>Conclusion 2 – Impact of New to the Firm Components (New IPs)</u>

Barclay and Dann (2000) considered product newness complexity, where newness increases the perceived complexity of developing the product. Clark (1989, p. 1260) concludes that "bringing parts engineering in-house and adding work by doing more unique parts design adds more engineering hours than one would expect from the amount of the increased workload," and that "the impact of scope on lead time works through changes in the difficulty of coordination in the planning process." Tatikonda and Rosenthal (2000) relate project complexity as the nature, quantity, and magnitude of organizational subtasks and subtask interactions required by a project. They consider the key determinants of complexity to be: the degree of interdependence among potential product/process technologies and the newness/ degree of difficulty of the project's objectives to the development organization. Tatikonda and Rosenthal's (2000) definition is powerful in that it focuses on complexity related to the nature of the work challenge posed by a project. Product novelty and newness imply an iterative design and development due to high levels of uncertainty. Organizations may not plan for an iterative development but in real-life the development will undergo numerous iterations and rework. Technological uncertainty relates to the uncertainty about different technological capabilities, best technologies to be used in the product and/or process, technical risks associated with different technologies, and the degree of familiarity of the team with the technologies involved in the project (Souder and Moneart 1992, Adler 1995, Olson et al 1995, Souder et al. 1998). When uncertainty exists, there is need for iteration as the product development may undergo many design changes. If uncertainty increases significantly, developers will have to carry out many iterations before a technical solution is found. Engineering changes orders (ECOs) occur at higher rates as the understanding of the technological capabilities is low and increases slowly over time (Murmann

1994). We have seen clearly that the new IPs caused more defects and rework compared to reused defects (refer to Figure 4)

<u>Conclusion 3- Moderating Behaviour of New and Reused Components</u>

*Fig 7 - MODEL 2 – PLOT OF IP Defects vs. NEW IPs / REUSED IPs*



It is very interesting that the increase in defects due to New IPs is moderated by the number of Reused IPs and similarly the increase in defects due to increase in Reused IPs is moderated by the number of new IPs. This relationship provides an opportunity to decide on the right combination of new and reused IPs to keep the productivity at manageable level. Figure 7 shows the moderating behavior of New and Reused IPs.

## SUMMARY

The results show that the system defects increases exponentially with the increase in total components. This clearly implies that the total development time will also increase exponentially due to the effort needed for fixing the defects uncovered. By adding more components to enhance functionality in fact results in escalating increase in cost due to increase in defects uncovered during the development time. The results indicate that there is an 8.5% increase in system defects due to unit increase in new component while 7% is the percentage increase in system defects due to unit increase in reused components. The results also indicate that the coefficient of the interaction term between new and reused components is statistically significant and negative. The reused components contribute to system defects to a lesser extent as compared to the new components as is to be expected given the fact that the reused components have been used before. Results provide detailed relationship between defects and new components and reused components and its impact on development time. This helps in deciding the optimal combination of new and reused components before starting the product development.

## REFERENCES

[1] Adler PS (1995). “Interdepartmental interdependence and coordination: the case of the design/manufacturing interface”. Organization Science, V 6(2), pp 147-167.

[2] Ali, A., 1994. “Pioneering versus incremental innovation: Review and research propositions”. The Journal of Product Innovation Management 11 (1), 46–61.

[3] Brooks, F. 1975. “The Mythical Man Month: Essays on Software Engineering”. Addison Wesley, Reading, MA.

[4] Browning, T.R., Eppinger, S.D., 2002. “Modeling impacts of process architecture on cost and schedule risk in product development”. IEEE Transactions on Engineering Management 49 (4), 428–442.

[5] Clark KB (1989). “Project scope and project performance: The effect of part strategy and supplier involvement on product development”. Management Science V 35, pp 1247-1263.

[6] Clark, K. B., T. Fujimoto 1991. “Product Development Performance: Strategy, Organization and Management in the World Auto Industry”. Harvard Business School Press, Cambridge, MA.

[7] Efron, B. 1979. Bootstrap Methods: Another Look at the Jackknife. Annals of Statistics 7:1—26.
[8] Evans, David H. (1963). "Modular Design—A Special Case in Nonlinear Programming". Operations Research 11(4):637–47.
[9] Griffin, A., 1997. "PDMA research on new product development practices: Updating trends and benchmarking best practices". The Journal of Product Innovation Management 14 (6), 429–458.
[10] Grupp, H., Maital, S., 2001. "Managing New Product Development and Innovation: A Microeconomic Toolbox". Edward Elgar, Cheltenham, UK.
[11] Hauser, J., Tellis, G.J., Griffin, A., 2006. "Research on innovation: A review and agenda for marketing science". Marketing Science 25 (6), 687–717.
[12] Henderson, R.M., Clark, K.B., 1990. "Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms". Administrative Science Quarterly 35 (1), 9–30.
[13] John Fox January 2002 – Bootstrapping Regression Models Appendix to An R and S-PLUS Companion to Applied Regression.
[14] Jurgen Mihm, Christoph Loch, Arnd Huchzermeier, "Problem-Solving Oscillations in Complex Engineering Projects". Management Science, 2003 - Vol. 46, No. 6, June 2003, pp. 733–750
[15] Kauffman, S. A. 1993. "The Origins of Order: Self Organization and Selection in Evolution". Oxford University Press, New York.
[16] Laudon, K. C., S. P. Laudon. 1991. Management Information Systems, 3rd ed. McMillian, New York.
[17] Leifer, R., McDermott, C.M., O'Connor, G.C., Peters, L.S., Rice, M., Veryzer, R.W., 2000. "Radical Innovation: How Mature Companies Can Outsmart Upstarts". Harvard Business School Press, Cambridge, MA.
[18] Levinthal, D. A. 1997. "Adaptation on rugged landscapes". Management Sci. 43(7) 934–950.
[19] McDermott, C.M., 1999. "Managing radical product development in large manufacturing firms: A longitudinal study". Journal of Operations Management 17 (6), 631–644.
[20] Morris P. W. G., G. H. Hugh. 1987. "The Anatomy of Major Projects". Wiley, Chichester, U.K.
[21] Murmann PA (1994). "Expected development time reductions in the German mechanical engineering industry". Journal of Product Innovation Management V 11, pp 236-252.
[22] Nichols, K., 1990. "Getting engineering changes under control". Journal of Engineering Design, 1 (1), 5–15.
[23] Nihal Orfi, Janis Terpenny, and Asli Sahin-Sariisik – "Harnessing Product Complexity: Step 1 – Establishing Product Complexity Dimensions and Indicators", The Engineering Economist, 56: 59–79, 2011
[24] Olson EM, Walker OC, and Reukert RW (1995). "Organising for effective new product development: The moderating role of product innovativeness". Journal of Marketing, V 59, pp 48-62.
[25] Otto, K. and Wood, K., 2001. Product design: techniques in reverse engineering, systematic design, and new product development. NewYork, USA: Prentice-Hall.
[26] Passy, U. (1970). "Modular Design: An Application of Structured Geometric Programming". Operations Research 18(3):441–53.
[27] Simon, H. A. 1969. The Sciences of the Artificial. MIT Press, Cambridge, MA.
[28] Souder WE and Moenaert RK (1992). "Integrating marketing and R&D project personnel within innovation projects: An information uncertainty model". Journal of Management Studies, V 29, pp 485-512.
[29] Souder WE, Sherman JD, and Davis-Cooper R (1998). "Environmental uncertainty, organisational integration, and new product development effectiveness: a test of contingency theory" Journal of Product Innovation Management, V 15, pp 520-533..
[30] Tatikonda, M. V., S. R. Rosenthal. 2000. "Technology novelty, project complexity and product development project execution success". IEEE Trans. Engrg. Management 47(1) 74–87.
[31] Terwiesch, C., C. H. Loch. 1999. "Managing the process of engineering change orders". J. Product Innovation Management 16(2) 160–172.
[32] Zhou, K.Z., Yim, C.K., Tse, D.K. –" The effects of strategic orientations on technology- and market-based breakthrough innovations". Journal of Marketing, Vol 69. Issue 2, April 2005