The Impact of a Requirements Specification on Software Defects and Quality Indicators

John Terzakis
Intel Corporation
e-mail: john.terzakis@intel.com

Abstract— **Capturing requirements in a written, centralized specification is a recognized product development best practice. But, how does this document impact software defect rates and quality? Is there any correlation between a well-written, properly reviewed requirements specification and software defect levels and other quality indicators? This paper will present data from an Intel case study illustrating the "before and after" scenario for a requirements specification. In the former, a minimal set of requirements were scattered across various documents for a first generation (older) product. In the latter, requirements were written and reviewed in a single requirements document for a second generation (newer) product. Software defect rates, feature commit vs. delivery, requirements volatility, and defect closure rates all improved dramatically even with the increased complexity of the newer product.**

Keywords-requirements specification; requirements defects; reviews; software defects; software quality.

I. INTRODUCTION

This case study involves two generations of a software product at Intel. The first generation product was developed without a requirements specification (e.g., Product Requirements Document or Software Requirements Specification). The requirements that existed were scattered across a variety of design documents, emails and web sites. There was no centralized source or repository for these requirements. The second generation product was developed based on a requirements specification. A standardized template was used along with a requirements management tool. Architecture specifications, design documents and test cases were developed from this specification. The requirements were rigorously reviewed by both technical content experts and a requirements Subject Matter Expert (SME). The second generation software product was more complex than the first in that the software had to run with, and implement functionality for, a next generation Intel processor. In addition, it had to combine code bases with a similar product from another business group.

II. DEFECT POTENTIAL COMPARISON

In general, there are many factors that impact the number and severity of software defects including: maturity of the team (development and validation), number of new features, complexity of the new features, test coverage and stability of the code base at the start of the project. In comparing the two software development efforts, the teams were of about equal size and maturity and their development methodology was the same (waterfall). The validation teams were also of similar size and maturity. There was some overlap of personnel between projects. As for a comparison of the two products, the newer product had more features, those features were more complex, the underlying hardware went through an architectural change, test coverage increased and the starting code base was less stable (due to the code merge from the other business group). Given all of these factors, the defect potential [1] should be higher for the second generation product than the first.

The most notable difference for the second generation product was the requirements specification. What impact would it have on overall software defect levels, quality, features delivered, number of change requests and defect closure rates?

III. REQUIREMENTS AND REVIEWS IN BOTH PROJECTS

Requirements for the first generation product were spread across documents, emails and web sites. That loose collection of requirements captured only about half of the initially intended product functionality. Reviews were held for those requirements that existed.

For the second generation product, the primary requirements author used a requirements management tool (RMT) to enter the requirements. The requirements were organized logically using key product features as section headers. The RMT had the capability to export to a document format. Reviews were based on this document.

We were the requirements SMEs assigned to work with the author to review and provide feedback on requirements quality. Initial requirements defect levels were high as this was the first set of requirements written by the author. However, with mentoring, peer reviews and stakeholder reviews, the requirements defect density for the requirements specification was reduced from about 4.75 defects per page in an initial revision to about 1.18 defects per page a later revision, a reduction of about 75%. The requirements specification became the basis for all architecture, design and test documents that followed.

IV. ACTUAL VALIDATION RESULTS

The following data presents a comparison of software defects, requirements volatility, feature variance and defect closure efficiency between the first generation ("Gen 1") and second generation ("Gen 2") software products.

 ISBN: 978-1-61208-202-8

Table I shows the total number of defects by type per product at the end of validation testing. Overall, the second generation product had about 50% fewer defects.

Table I: Total Number of SW Defects

Defect Type	Gen 1	Gen 2	Delta
Critical	21	3	-86%
High	137	69	-50%
Medium	111	62	-44%
Low	24	6	-75%
Totals:	293	140	-52%

Table II shows the requirements volatility per product at key milestones during development. Some requirements volatility is due to scope creep (requests for new features) but most of it is due to changes needed due to missing, incomplete or incorrect requirements. At release, the second generation product had almost half the volatility of the first generation.

Table II: Requirements Volatility at Major Milestones

Milestone	Gen 1	Gen 2	Delta
Alpha	0.4	0.4	0%
Beta	1.2	0.7	-42%
Release	1.7	0.9	-47%

$$\text{Volatility} = \frac{\text{\# of added+changed+deleted requirements}}{\text{Total \# of requirements}}$$

Table III shows the feature variance per product at key milestones during development. This metric shows how well the features delivered in final product matched what was committed by the team to be delivered. The second generation product was able to deliver many more features than the first generation product at release.

Table III: Feature Variance at Major Milestones

Milestone	Gen 1	Gen 2	Delta
Alpha	0.05	0.15	+300%
Beta	0.15	0.25	+167%
Release	0.15	0.35	+233%

$$\text{Feature Variance} = \frac{\text{(Current - Planned Features)}}{\text{Planned Features}}$$

Finally, software defect closure efficiency (cumulative SW defects closed / cumulative SW defects submitted) at the end of validation testing improved from about 69% in the first generation product to about 87% in the second generation product, an improvement of over 25%. Note that a higher percentage indicates that defects are being closed more rapidly. This means the development and validation teams are spending less time identifying, researching and correcting software defects.

V. CONCLUSIONS

A number of factors could have had some impact in reducing the number of software defects from the first to the second generation product. They include applying lessons learned from the first development to the second, augmented developer experience and maturity, improved code review practices and more rigorous unit testing prior to the start of validation. No doubt these factors had some influence on improving software defect levels. However, given the increased complexity of the second generation product, they should have had a minimal effect on total software defect density levels. The key software quality indicators showed a dramatic improvement in the second generation product. Some other factor was playing a dominant role in these improvements.

Clearly, a well-written, properly reviewed requirements specification was the major contributing factor to these improvements in software defects and other quality indicators on the second generation product. This set of requirements had a positive influence on the total number of software defects (down 50%), requirements volatility (down 50%), feature variance (improved 2x) and software defect closure rates (improved by 25%). A third generation product is currently in development. Results from that project will be analyzed in a future paper.

REFERENCES

[1] Jones, Capers, *Software Quality: Analysis and Guidelines for Success*, International Thomson Computer Press, June 14, 2000.