A Study on the Perceived Value of Software Quality Assurance at JPL

Dan Port
University of Hawaii
dport@hawaii.edu

Joel Wilf
Jet Propulsion Laboratory
Joel.Wilf@jpl.nasa.gov

Abstract
As software quality assurance (SQA) moves from being a compliance-driven activity to one driven by value, it is important that all stakeholders involved in a software development project have a clear understanding of how SQA contributes to their efforts. However, a recent study at JPL indicates that different groups of stakeholders have significantly different perceptions about the value of SQA activities and their expectations of what constitutes SQA success. This lack of a common understanding of value has fragmented SQA efforts. Activities are driven by the desires of whichever group of stakeholders happens to hold the greatest influence at the moment leading the project as a whole not realizing the full or needed value of SQA. We examine this and other results of the recent study and how these impact both the real and the perceived value of SQA.

1. Introduction
This study reports on stakeholder perceptions regarding the value of software quality assurance (SQA) currently practiced by the NASA Jet Propulsion Laboratory (JPL) Process and Product Assurance group. We examine where these perceptions conflict with each other, as well as the consequences of this conflict and the resulting confusion among stakeholders about SQA.

The results in this paper stem from interviews and surveys with over 40 SQA stakeholders, analysis of definitions and descriptions of SQA in JPL documents, and direct observations of SQA as it is practiced at JPL. The critical stakeholders identified in this study include:

Table 1: Critical SQA Stakeholder Groups

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>SQA Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission Assurance Managers (MAM)</td>
<td>Acquirer</td>
</tr>
<tr>
<td>Flight Software Managers (FSM)</td>
<td>Customer</td>
</tr>
<tr>
<td>Project Managers (PM)</td>
<td>Customer</td>
</tr>
<tr>
<td>Software Quality Improvement staff (SQI)</td>
<td>Assessor</td>
</tr>
</tbody>
</table>

While there are a number of additional important stakeholder groups, we will generally only focus on those listed in Table 1, since they are fundamental to the SQA value issue. Our study strongly suggests that there is confusion regarding the perceived value of SQA across all stakeholder groups. A clear indicator of this is seen in Figure 1, which shows the overall stakeholder responses to the following statements in our SQA issues survey: (D) “Managers have trouble rationalizing SQA (cost justification, benefits, value, degree required, etc.) and (I) “There is limited understanding or confidence in how SQA reduces uncertainty in system and project quality.” The boxplots in Figure 1 show that all stakeholders (ALL) listed in Table 1 tended to “agree” or “strongly agree” with these assertions. Note: we present all survey results as standard boxplots which visually represent the 5-number summary where the “box” represents the 25th to 75th percentile of all responses, the thick vertical line within the box represents the median, and a “whisker” vertical line connected by a dashed line to the box represents a minimum or maximum, except for a possible outliers, which are represented by dots. See [7] for details.

Figure 1: Survey Response Summary, Questions (D) and (I) From ALL Stakeholders

While this study was conducted only at JPL, our results have been presented at a recent NASA Software Assurance workshop, where the ensuing discussions led us to conclude that confusion over SQA value is
pervasive across all NASA centers and likely to a large degree among Aerospace contractors.

Many factors contribute to conflicting stakeholder views of SQA value, but we find that fundamentally the current practice of SQA does not clearly connect SQA activities to project value (positive outcomes for the project). The problem is systemic and pervasive, and not limited to a given project. Consequently it is becoming increasingly difficult to determine and justify how much to invest in SQA and to assess whether SQA has been performed adequately on any given project. Thus, one of the main objectives of this work is to comprehensively uncover possible sources of confusion regarding the value of SQA.

It is important to keep in mind that while we conclude that currently there is a confused perception of SQA value, this does not imply that no value exists or even that there is a perception that SQA is valueless. To the contrary, our study results imply that the majority of stakeholders do believe SQA offers value, and furthermore there is objective evidence that some of this value is being realized. However, the confusion about SQA value does have serious consequences. For example, a project manager may skip an SQA delivery review in favor of additional component testing due to confusion about how such a review can help decide whether a delivery is ready for integration testing. This manager may feel that additional component testing is the only tangible means of reducing such doubts.

The results presented here are based on subjective data from stakeholder surveys, supplemented with objective data inferred from documents and observations of practice. An important basic presumption we make in this work is that perception drives practice – and that both perception and practice are continually evolving at JPL. Therefore, the results described here represent a snapshot of current practice and a bit of history, based on small samples taken over a window of time. Not all results will be true all the time for all situations forevermore.

2. Overview of SQA

Over time, JPL has increasingly recognized the importance of software quality. This is in part a response to failures due to software on missions such as the Mars Climate Orbiter and Mars Polar Lander. Concerns are compounded by the increasing size, complexity, and cost of software development at JPL. Indeed, as seen in Figure 2, the historical increase in code size as measured in “standardized” source lines of code1 (SLOC) is exponential.

Projects such as the Mars Science Laboratory (MSL), scheduled to launch in 2011, is projected to have nearly 1900K SLOC, so there is no indication that the trend shown in Figure 2 is unrepresentative. Data collected by Capers Jones [4], ISBSG [2] and others have established that the number of defects2 is strongly positively correlated with size as indicated in Figure 3 (note that this data shows overall defect density with respect to size and does not indicate anything about historical defect density changes over time). Consequently, increasing code size implies an increasing expected number of defects. Furthermore, under such size considerations, testing alone, no matter how extensive, is able to find and remove all defects [5],[6].

---

1 Where a line of code is regularized to eliminate code formatting and style differences, exclude code comments, etc.

2 Defects here include an error in any artifact (e.g. requirements) and not limited to bugs in code
The conclusion of this from an assurance perspective is that software risk (generally measured by the likelihoods and consequences of failures) is also increasing. To be sure, Figure 4 shows that MAMs view SQA as a critical area of mission assurance.

The question is: what specific part can SQA play – relative to other techniques, such as automated testing – in improving this situation? One can evaluate the costs and benefits of different investments in software quality. Then, based on economic arguments, a decision is made on where and how to improve quality. The required level of quality for a particular project is not an inherent attribute of the system, but rather it is a business decision [1].

To make that decision, management needs to determine how much to invest in quality and what benefits to expect. Management also needs to compare alternative investment strategies to determine which one to focus on. SQA contributes to defect avoidance and early defect mitigation, increasing productivity and reducing both re-work and maintenance costs. Also, relative to testing, quality assurance is highly cost-efficient in lowering the risk from residual defects. One would expect that under these circumstances investment in SQA activities would be increasing.

Yet the size of the SQA group has actually gotten smaller in last five years, indicating that the institution is investing less in SQA services relative to increasing size and complexity.

We do not have data comparing investment in SQA relative to related activities such as testing, but Figure 5 from our survey of direct project “customers” (FSMs) indicates that these activities tend to be more valued than SQA.

Figure 4: Survey response by MAMs to question (17) “SQA is a critical area of mission assurance”

![Survey response by MAMs](image)

Figure 5: Survey Response From FSMs Question (1) “It is better to invest more in testing than perform SQA” and From PMs Question (21) “It is better to invest in more V&V than SQA”

We have found that the decrease in SQA investment – reasonable or not – is firmly based on a declining and confused perception in the value of SQA.

### 3. Definitions and Expectations of SQA

Currently stakeholders have a wide range of viewpoints on the nature and practice of SQA. In particular, stakeholders have differing views on which activities are within the scope of SQA practice, what results are expected, and what constitutes SQA “value.” The next section lays out some of these different viewpoints and how they come in conflict with the perceived value of SQA.

#### 3.1. Definitions of SQA

The value disconnect begins and is rooted in the existence of multiple definitions of SQA, which can be found in various official documents. These definitions are somewhat conflicting, which lead to conflicts in stakeholder expectations, obscuring the SQA value proposition. Consider for example, the difference between the NASA definition of Software Assurance, the definition in the JPL SQA Charter, and one taken from the ICCA Mission Assurance Plan:

**NASA definition:**
The planned and systematic set of activities that ensure that software life cycle processes and products conform to requirements, standards, and procedures [NASA-STD-8739.8]

**JPL charter:**
The Software Quality Assurance (SQA) Group promotes awareness of and reduces exposure to software-related risk for JPL missions.

**ICCA Mission Assurance Plan:**
The purpose of SQA is to achieve the highest quality-to-cost ratio within the Project’s constraints

---

Footnote:

3 Failures are the results of *residual defects* or a defect missed in testing, that shows up in mission operations
and policies and to increase the probability of overall mission success.

The advantage of the latter definition is that it clearly states a value proposition. Generally speaking, though, the definitions do not connect the implied or stated activities to value objectives (even when these are stated). Hence it is not surprising that these connections are not made in practice when the above definitions are used to plan SQA activities.

**SQA as Early Defect Detection:**

One prevalent definition of SQA originates in the cost-to-fix detection/removal model (defects are much less costly to fix early in the development life cycle). Informally this defines SQA as those activities that directly contribute to the identification of defects earlier than would be found in testing and reviews. The value proposition is that early defect identification leads to the removal of defects sooner in the development, where it is presumably less expensive, avoiding costly late development rework, less re-testing, and lower risk of costly operational defects (faults that lead to failures) slipping through reviews and tests.

**3.2. Stakeholder Expectations of SQA**

Generally different stakeholders subscribe to different definitions of SQA. A common example of this arises when one stakeholder’s definition implies an expectation of active assurance “intervention” while another stakeholder’s definition expects more passive “investigative” assurance. Intervention assurance activities aim to directly ensure quality through process recommendations or proactive problem detection or avoidance, before an artifact exists or while it is being developed (e.g. recommendations on what tests to perform). Investigatory assurance activities are quality assessments of existing artifacts, reporting issues on existing artifacts and processes (e.g. requirements traces to test cases). It is a well-established premise of software assurance that there is no expectation that assurance activities directly change the quality of the software product or process (for example see [2]). Yet clearly if a stakeholder expects SQA to “tell me what to do to avoid trouble”, they are likely not going to get value from “tell me what might be trouble” kinds of SQA activities and vice versa.

That there are notably different expectations between SQA stakeholders is apparent from the contrasting responses to survey question (A) “The specific goals, objectives, and expectations for SQA are not clearly defined for a given engagement” between SQA “planner” and “provider” stakeholders (MAMs and SQE’s) and SQA “consumers” and “customers” (FMS’s and PM’s). In Figure 6 it is apparent that the consumers of SQA are confused on the expected outcomes of an SQA engagement, whereas the SQE’s feel that this is clear, and MAMs show a mixed response, tending towards confusion.

Interviews reveal the cause of this discrepancy is stakeholders’ use of different definitions leading to different objectives for SQA. Project managers, for example, would define SQA in interviews similar to “assuring the delivery of high quality Flight Software that will achieve mission objectives” [from the GRAIL Project Software Management Plan]. Yet the SQA group would say their objective was to “reduce software risk”. While these views are not necessarily incompatible, the differences cause confusion.

**Figure 6: SQE, MAM, FSM, PM Responses to Survey Question (A)**

Surprisingly our interviews with stakeholders indicate that the expected outcomes of successful SQA are reasonably clear and generally accepted. These are:

1. **During operations:** fewer missions failing due to software quality problems
2. **During development:** confidence that code and other work products have an appropriate level of quality throughout the life cycle, especially at “gateway” decision points such as acceptance reviews
3. **Support for decision-making:** reduce risk of making a “bad” decision.

However, this consistency is at a level that is too general to be used as “practical” success criteria for the outcome of an SQA engagement. Consequently, each stakeholder interprets the above as a variety of different “win conditions” that address their particular SQA success needs.

Table 2 shows the win conditions we have observed for various SQA stakeholders at JPL. As the table reveals,
there is great variety in the win-conditions for different stakeholders.

Table 2: SQA Stakeholders and Their Win Conditions

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>Win Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project managers</td>
<td>Ensure complete compliance (“large set of JPL pants”)[2]</td>
</tr>
<tr>
<td>Software element managers</td>
<td>Discover agile problems (delay to reach product)[3]</td>
</tr>
<tr>
<td>CQF</td>
<td>Certify any SW for JPL ATL10 flight (change of approval)[4,5]</td>
</tr>
<tr>
<td>MAM</td>
<td>Accredited SW risk[6]</td>
</tr>
<tr>
<td>FSM</td>
<td>Ensure testing a current and complete[2]</td>
</tr>
<tr>
<td>SQM</td>
<td>Ensure compliance[7,8]</td>
</tr>
<tr>
<td>PM</td>
<td>Ensure a quality of SW proactive improvements[1,2][9,10]</td>
</tr>
<tr>
<td>SQIs</td>
<td>Ensure SW quality[11,12]</td>
</tr>
</tbody>
</table>

The predicament here is that to “meet expectations” for SQA presently translates into “satisfy all stakeholder win conditions.” But Table 2 shows how hard it is to achieve this. The win-conditions vary among the stakeholders, they may not be mutually compatible or consistent, and even if they were, it is doubtful that there will be enough funding or time to adequately address all stakeholder win-conditions.

The SQE’s have evolved a variety of strategies to deal with this predicament. These include:

- Focus on the “most important” stakeholder(s)
- Negotiate with or pressure the stakeholders to agree on a smaller set of win conditions.
- Satisfy win conditions only for selected areas or subsets
- Do as much as possible then ask for liens until complete.

At times an SQE will perform activities outside the SQA charter to meet stakeholder particular expectations (see 7 Response to (B) “SQE’s often provide services outside the scope of the SQA charter e.g. testing”). Generally though, all these strategies are flawed. Any approach that allows one stakeholder to “win” and another stakeholder to “lose” will eventually end up with all stakeholders losing. Accordingly, expectations tend to go unmet, resulting in frustration and further perplexity on the value of SQA.

A common understanding is defect density, but alternatively we have observed the following representations for software quality:

- Number of critical defects / faults / failures (FMECA view)
- “ilities” – quality attributes (e.g. reliability, dependability, maintainability, etc.)
- Testing until a threshold is passed
- Process compliance
- Risk level

The above understandings are commonly held, yet differ from the “official” JPL definitions of quality, which are from a Verification and Validation (V&V) perspective:

Official 1: Quality is fulfillment of requirements (shown by Verification)
Official 2: Quality is “fitness for use” (shown by Validation)

4. SQA Perspectives that confuse value

We have described the confused state of SQA with its complex collection of stakeholder expectations, working under multiple and likely incompatible definitions. Here we elaborate on the many different perspectives of what SQA is or should be at JPL. We discuss these not because any of these perspectives are necessarily wrong, rather that we wish to point out various value related issues for each perspective.

Different perspectives, different goals

Various perspectives on SQA have shaped a diversity of goals for software assurance. “Officially” the SQA charter states the goals are:

- To lower software risk
- Ensure compliance to institutional standards and requirements (e.g. SDSP)
- Ensure compliance to standards and processes adopted by project (e.g. SDSP tailoring)

However these are frequently superseded by other stakeholder’s goals. For example, sometimes a project manager just wants SQA to review key documents or certify software deliveries. The Software Quality Improvement (SQI) organization has a primary goal to perform PPQA to meet CMMI Maturity Level 3 certification. While these goals are not necessarily in conflict, they contribute to the confusion of the value of SQA.
**SQA as testing**

One view of SQA is that of extended or alternative testing. The difficulty with this view is that SQA test findings tend to be treated as defects that aren’t as important as bugs identified by developer testing. Furthermore, often in these situations SQA is used to test work products for which no “real” tests exist or have not been specified. These both miss the value of software assurance.

**SQA as Verification and Validation (V&V)**

NASA includes V&V and Independent V&V (IV&V) as assurance disciplines, and as such they have much in common with SQA. Often both V&V and SQA are performed within a project and there is confusion on the boundaries between V&V (a systems engineering practice) and SQA (an assurance practice). Because V&V and SQA share many of the same activities (e.g. requirements inspection), one might expect some stakeholders to say that V&V can replace SQA. However, SQA stakeholders, in particular project managers, viewed SQA’s independent perspective as valuable. Their response to survey question (20) “SQA should be handled by the developers themselves” was entirely “disagree” to “strongly disagree.” Nearly all PMs interviewed commented on the value of an independent perspective in response to survey question (20):

- “Always want an independent perspective”
- “Need independent review – developers too close”
- “Since the SQE is if a different organization there is value of an independent check”
- “… need independent effort. For requirements verification etc. need some, not necessarily SQA, independent (unbiased) part, but if SQA is the most cost-effective choice…”

Given this, it’s not surprising that some stakeholders view SQA as a kind of independent V&V (IV&V). However, taking this contradicts a main tenet of IV&V – compete independence. SQA’s has an embedded relationship with the project, being both funded by and actively ensuring (in principle) the quality of the software. Also, the “customers” for SQA are the project and the institution, whereas for IV&V the customer is NASA. Trying to substitute one for the other or “double up” only serves to confuse the differing value propositions of both.

**SQA as just another set of reviews**

There are some stakeholders who view SQA as another excuse for (perhaps superfluous) reviews. The trouble with this perspective is that SQA is not explicitly integrated with the standard review life cycles. Moreover most SQA activities do not follow a project through their “gateways” to the goals. This view is counter productive as project “review fatigue” dampens desire to cooperate with SQA activities.

**Assurance for insurance**

Another confusing perspective on SQA is as an “insurance policy” against blame because “it was checked that we did all the right things.” So SQA helps project avoid trouble with the JPL and NASA “compliance police” and providing a “due-diligence” defense in case of failure. We note that due-diligence is often valued only after a problem occurs. Hence the drawback with this view is that it rather than reducing the risk of problems, the aim is to pass responsibility and blame elsewhere.

**Assurance as risk management**

The SQA charter says assurance is performed to lower software risk. The problem with this view is in understanding exactly how SQA activities lower software risk. Some stakeholders believe that assuring compliance is a reasonable proxy for risk reduction. The FSMs response to survey question (10) “Compliance to process is important for developing high quality software” was entirely “agree” and “strongly agree”. However, this is accepted without empirical evidence, on the basis that institutional requirements, policies, and design principles are derived from “best practices.” However there is skepticism that these universally apply to every project. Risk management through assuring compliance, as practiced currently, has unclear impact on actual project risk and hence difficult to make the basis of SQA value.

**Assurance as quality management**

There’s a view that SQA should help projects avoid and repair quality problems. SQA is viewed as a source of solutions – “don’t just tell us what’s wrong, fix it or tell us how to fix it.” There is an expectation that SQA is responsible for setting or achieving quality goals (e.g., “ilities”) and failure to do so is a failure of SQA. However this implies SQA is fully empowered to take action on quality issues. In actuality, SQA is limited to providing information to the project through the reporting findings and making recommendations. Were this not to be the case and SQA could directly manage quality on the project, SQA would lose its value as an independent voice on the project.

**5. Damaging SQA misperceptions**

There are a number of misperceptions that degrade the perception of SQA value. These misperceptions are not universally held, but are prevalent enough to
perpetuate confusion over the value of SQA. We focus on five harmful misperceptions:

1. SQA is not needed
   Some developers believe that SQA is not needed because JPL software has been successful in the past without SQA involvement. Despite recent software failures and numerous software quality related mishaps, there remains a pervasive “hero” culture that depends on exceptionally talented developers and the belief that thorough testing will fix the rest of the problems. Many of these same people also believe that SQA doesn’t matter unless they find something significantly wrong that has not been identified elsewhere (Figure ). Some managers may also think SQA does not change quality, since it is not part of decision-making, and it’s not on the critical path to achieving project goals (Figure 17). Furthermore, some believe that if software quality issues arise, they can always patch the system post-launch. As long as the system can boot up and software patches can be uploaded, there is no need for software assurance. This is paraphrased from a well-known project manager who called this approach “an acceptable strategy for managing schedule slippage and software quality.”

2. The defect of focusing only on defects
   Some stakeholders believe that SQA should focus only on the defects not found by other methods. But as discussed previously, assurance objectives focus on the quality of the software as a whole. This view doesn’t address completeness of requirements, sufficient risk management, and adequacy of other development artifacts and factors. This view fails to make quality estimates such as the quantity and severity of defects remaining in the system. Hence there is no basis for establishing whether the system quality is “good enough” or needs further quality improvement. This uncertainty confounds critical development decisions based solely on defects found.

3. SQA can be done by anyone
   Another common damaging misperception is that SQEs are technically inferior to developers and bring to bear no specialized knowledge, training, or skill. Those who believe this also tend to think that developers can achieve quality goals themselves with their own testing and advanced tools such as static code analyzers like Coverity [3].

4. SQA is only of value in helping a project pass reviews
   While generally true that stakeholders value SQA for assisting in passing reviews (see Figure 8 response by ALL stakeholders to question (J) “A focus on assisting projects to pass milestone reviews has been successful and of value (proactive involvement is especially appreciated”), exclusive focus on this places various stakeholders in conflict. MAMs appreciate the credibility for review milestones and requirements that SQA provides, PMs seem to appreciate SQE facilitation for preparing and passing reviews (see Figure 8 PMs response to question (24) “SQA is valuable to prepare and pass reviews”). However, FSMs do not believe passing reviews should be the main focus of SQA (see Figure 8 for FSMs response to question (3) SQA should focus mainly on helping a project pass reviews).

5. Confusion of SQI and SQA
   Figure 9 shows that there is a significant subset of stakeholders who are confused about the difference between SQA (software assurance) and SQI (process improvement) efforts and initiatives. In our interviews some stakeholders expressed concern about a redundancy between these two separate groups. Particularly troubling is that MAMs and SQI personnel generally felt there is more confusion than other stakeholder groups.

6. Consequences of Value Confusion
   The confusion of the value issues described above present a number of challenges to the effective practice and vitality of SQA. These challenges impact both the real and the perceived value of SQA.
Difficulty establishing a working definition of SQA
Confusion about SQA value has made it difficult to achieve complete success in planning, managing, assessing, and improving SQA practice at JPL. Without a working definition, SQA planning appears ad-hoc, leaving doubt about what assurance is expected to achieve for a particular project. It has only been in the last few years, that standard JPL SQA requirements and processes have been developed and used.

The mixed response to survey question (G) in Figure 10 suggests that there is increasing awareness of an institutionalized SQA process whereas previous surveys have shown this to be absent. The PMs response to survey question (G) “SQA is planned in an ad-hoc manner and differently for each SQE (no institutionalized process)” was consistently “do not agree” providing further evidence of progress. One PM commented, “Historically this [survey question (G)] was true, but we seem to have one [an SQA process] now.”

Figure 10: Response by stakeholders to Survey Question (G)

SQA presented as “grab-bag” of activities
Typically an SQA engagement begins by negotiating a set of activities taken from a “menu” of activities such as listed in the NASA Software Assurance Standard. Presumably activities are chosen with respect to what assurance value they provide to project, but it is unclear what basis is used to make these choices. Often the driving factor is institutional compliance. Even so, there is no clear guidance as to whether the activities are able to fulfill their objectives, to what degree the activities are needed, and if the aggregate set of activities can realize assurance goals.

No guidance on valuing of SQA activities
Currently JPL SQA is guided by institutional requirements, rather than a valuation of SQA activities. This “value neutral” approach does not account for the inevitable variations in assurance concerns from project to project. The resulting perception of SQA is that it favors institutional over project concerns. Figure 11 indicates that currently the perception is that SQA effort is oriented more towards compliance rather than the value of that compliance to the given project.

Figure 11: Responses from each stakeholder to the statement “SQA assurance fulfills institutional rather than project needs”

Reporting Findings but Not Software Quality
Assurance is having greater certainty as to the quality of the system and the processes used to develop it. Reporting both quality problems (negative findings) and verifying that acceptable quality is present (positive findings) are equally important from the assurance perspective. Without value-based SQA, the emphasis tends to be overwhelmingly on negative findings, rather than positive ones. As indicated in Figure 12, positive findings indeed have value to FSMs.

Figure 12: FSM Response to Survey Question (9) “There is no value in an SQA finding that indicates nothing is wrong”

Figure 13: FSM Response to Survey Question (10) “SQA is only useful if it can uncover non-trivial defects we did not already know about”

Adding to this, there are some who the number of new, non-trivial findings as the measure of SQA success. This is shown in Figure 13. Unfortunately, the pressure to produce a large number of new findings can result in
too many trivial ones. This is what may be reflected in Figure 14, which shows that SQA is sometimes criticized for having too many “trivial” findings.

Connection to risk unclear
The value of SQA is necessarily tied to how the results of assurance activities (or “findings”) ultimately impact cost, schedule, function, and system quality risks. Presumably assurance goals are expressed with respect to managing such risks. Yet clear and measurable connections of assurance findings such risks are in generally not expressed or known. As Figure 15 shows, there is a high dependence on “expert” (e.g. MAMs) tacit knowledge and experience to judge achievement of assurance risk management goals. This subjective ad-hoc approach inhibits true value based planning of SQA activities and complicates communicating said value to stakeholders.

Risk assurance not performed
Risk assessment is a vital and fundamental assurance consideration. In essence, the intent of software quality assurance is to increase the confidence in the understanding of risks. Primarily this is accomplished through reducing the uncertainty regarding the qualities of the product and development possesses. Yet as we have seen in the survey response to question (I) in Figure 1, it is unclear to what degree SQA activities address the assurance of risk. This is important as developers acknowledge that there is notable risk with respect to software quality and more notably, with architecture choices (see Figure 17).

Rationale for scope lacking
A major source of value degradation lies within the insufficient manner in which SQA is planned and concluded. For planning, rationale for how much SQA needs to be performed is rarely provided (other than “all finding for SQA is used”). Similarly rationale is not given as to why enough SQA had been performed and no further SQA is required (other than requirements have been complied with). Without such
rationale stakeholders may be uneasy about the scope of an SQA engagement. Such uneasiness is evident in the conflict between the SQE’s and other stakeholders on how the scope of SQA is determined (see Figure 18).

![Figure 18: Comparison of SQE with Other Stakeholder’s Response to Survey Question (K) “It is not known how to determine how much SQA to do and if enough SQA has been done (other than ‘no funding left’)”](image)

**Priorities unclear**
There is confusion on the prioritization of SQA activities and which areas to assure. The value of SQA cannot be realized if important findings are not addressed. As indicated in Figure 19, it is common that findings go unaddressed.

![Figure 19: FSMs Response to Survey Question (5) “There is often not enough time or budget to address all SQA findings.”](image)

Given the highly constrained conditions (cost, schedule, access to artifacts etc.), complete assurance is not practical from both the SQE and developer’s ability to respond to findings. Consequently, objective and meaningful prioritization is critical to successfully realizing value from an SQA engagement. However as practiced currently, SQA activities are not always prioritized meaningfully (Figure 20). The activities and work agreement appear to be arbitrarily negotiated and not explicated directed towards achieving assurance goals and objectives.

7. Summary and Conclusions

We have presented a number of results from a recent extensive study investigating the perceptions of value of SQA at JPL. These results make it clear that these perceptions create challenges that drive the practice. In recent years, SQA practice at JPL has transitioned from an ad-hoc tradition-based practice to a requirements-driven practice, but confusion over the value of SQA remains. Investigation into the sources of confusion and its consequences is driving JPL towards defining a value-based practice whose aim is directly address value issues and improve the real and perceived value of SQA.

![Figure 20: Response by SQE, MAM, FSM, PM, SQI Stakeholders to Question (C) “SQA activities tend not to be prioritized meaningfully (i.e. “best judgment”)”](image)

8. References