# S3106: PBRA and RBA Process

**Version: D**

**Effective Date: October 4, 2012**

Note:  The official version of this document is maintained in IV&V's internal IV&V Management System Website (https://confluence.ivv.nasa.gov:8445/display/IMS).  This document is uncontrolled when printed.



**Portfolio Based Risk Assessment (PBRA)**

**Risk Matrix to Support the IV&V Program Portfolio**

**And**

**Risk Based Assessment (RBA)**

**Risk Based Assessment for NASA IV&V Projects**


PBRA Credits:

The PBRA process was initially defined by Steve Driskell, Marcus Fisher, Tom Marshall,

and Kurt Woodham

It has been further refined based upon comments and suggestions from users.

RBA Credits:

The RBA process was initially formulated by an assessment team supported by Anita Berns,

Ken Costello, Darilyn Dunkerly, Dan McCaugherty, Christina Moats, and Harry St. John.


Details of the RBA assessment can be found here:  Livelink/ECMLES (online)/Enterprise Workspace/IV&V OFFICE/TQ&E/Process Asset Assessments/RBA Assessment – 2010 (https://ecmles.faircon.net/livelink/livelink/Open/1160964)

Current Process Owner:

Technical Quality and Excellence (TQ&E) Lead

## *Introduction*

IV&V, as a part of Software Assurance, plays a role in the overall NASA software risk mitigation strategy applied throughout the lifecycle, to improve the safety and quality of software systems.[1] In order to understand the software risk profile within NASA, NASA IV&V performs assessments of risk on Mission Projects. These assessments are intended to meet two objectives: 1) to create a portfolio to support prioritization of technical scope across all IV&V projects, and 2) to create a mission-specific view to support planning and scoping of NASA IV&V Project work on each individual IV&V Project. This document contains a two phase process that supports both of these objectives. Phase One, which supports objective 1, is known as Portfolio Based Risk Assessment (PBRA). Phase Two, which supports objective 2, is known as Risk Based Assessment (RBA).

PBRA results in a risk score for each high level capability for a particular mission. RBA results in a risk score for each system/software entity for a particular mission. RBA will likely be performed iteratively during the IV&V Project lifecycle, as additional information about the mission and software becomes available.

### Definitions

- Capability – the action or reaction of the system desired to satisfy a mission objective; what the system must be capable of doing in order to satisfy mission objectives.
  Limitation – a constraint or condition that can keep a desired action or reaction of the system from occurring, or that can keep a desired action or reaction from occurring in its entirety
    - Results of IV&V provide evidence of limitations in a system's capabilities.
- Relative importance weight – a factor applied to the final risk score *after* the risk assessment. It is derived from the software inventory and is used to differentiate among capabilities that share the same risk score.
- Three Questions – Questions 1, 2, and 3 are identified below:

    1. Will the system's software do what it is supposed to do?
    2. Will the system's software not do what it is not supposed to do?
    3. Will the system's software respond as expected under adverse conditions?

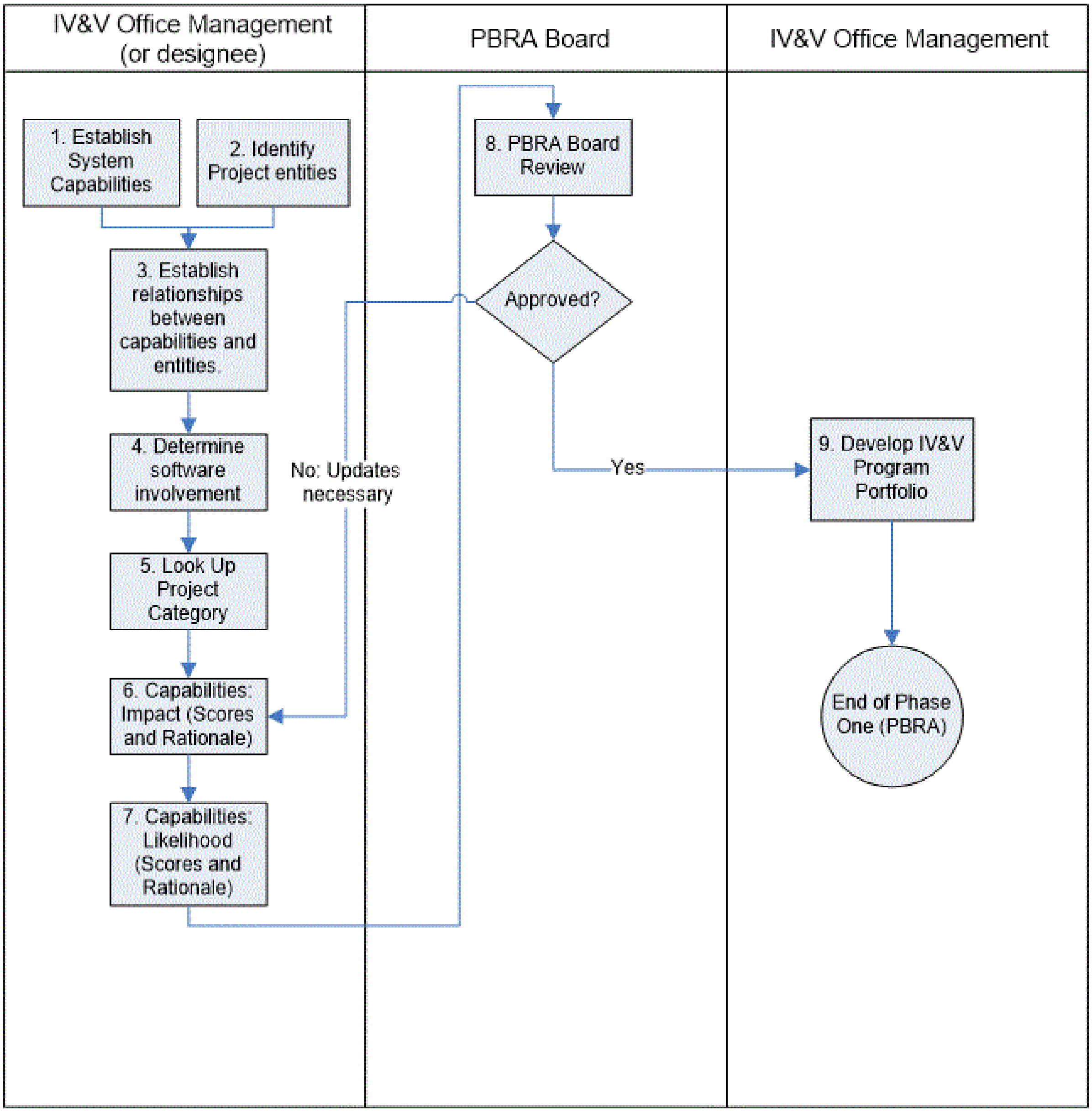
### Acronyms

| | |
|---|---|
| AMPL | Agency Mission Directorate Program and Project List |

| | |
|---|---|
| APXS | Alpha Particle X-Ray Spectrometer |
| C&DH | Command and Data Handling |
| COTS | Commercial Off The Shelf |
| DAN | Dynamic Albedo of Neutrons |
| EDL | Entry Descent and Landing |
| GOTS | Government Off The Shelf |
| GNC | Guidance Navigation and Control |
| IBA | IV&V Board of Advisors |
| IF | Interface |
| MAHLI | Mars Hand Lens Imager |
| MARDI | Mars Descent Imager |
| OSC | Operational Software Control |
| P | Performance |
| PBRA | Portfolio Based Risk Assessment |
| PCF | Project Category Factor |
| PS | Personnel Safety |
| RAD | Radiation Assessment Detector |
| RBA | Risk Based Assessment |
| REMS | Rover Environmental Monitoring Station |
| SAM | Sample Analysis at Mars |
| SA/SPaH | Sample Acquisition, Processing, and Handling |
| TS&R | Technical Scope and Rigor |

***<u>Phase One:  Portfolio Based Risk Assessment (PBRA)</u>***

Figure 1 depicts the PBRA process, which results in establishment of the IV&V Program portfolio.




**S3106 Phase One PBRA -- 05-15-2017.vsdx**

**Figure 1: Approach for establishing IV&V Portfolio using a risk-based assessment approach.**

## Process Steps

Note #1: Steps #1 and #2 can be performed in parallel.

Note #2: Steps 1 through 7 are performed for each IV&V Project.

Note #3: IV&V Office Management is responsible for the entire PBRA process, but may choose to delegate Steps 1 through 7 as appropriate.

Note #4:  The output of this process will need to be periodically revisited for active IV&V Projects (for example, to support IV&V Board of Advisors (IBA) needs).

1*. Establish the system capabilities, which represent the desired behaviors of the system to satisfy the goals of the mission and establish the context for the system's software*.

a. An example capabilities list is in the following table (the information in the table should be accompanied by additional content, including a description of each behavior).  In order to fully understand the collective risk within the mission, it is necessary to perform this elaboration down to the segment level (shown below) and potentially further.  Decomposing the system capabilities into their respective segment capabilities provides more information that will make the risk assessment less subjective — meaning that, in assessing the Impact (the Performance Category, to be specific) the assessor also evaluates the system capability as it relates to the mission objectives.  If the system capability is broken down into its segment capabilities, then the segment capabilities can be evaluated as they relate to the system capability, and so on.  Information  can be taken into consideration at these lower levels  to lessen the amount of subjectivity in the assessment.
b. Scoring and rationale for system capabilities (e.g. Launch to Mars, Cruise to Mars, Maintain flight systems, etc.) is the required output of the PBRA process.
c. Example Capabilities:

| | | | |
|---|---|---|---|
| Conduct habitability investigations | | | |
| | Launch to Mars | | |
| | Cruise to Mars | | |
| | | Trajectory control | |
| | | Attitude Control | |
| | Approach Mars | | |
| | | Trajectory control | |
| | | Attitude Control | |
| | Maintain flight systems | | |
| | | Establish and maintain power | |
| | | Establish and maintain thermal control | |
| | | Perform fault detection | |
| | | Establish and maintain communications | |
| | | Gather engineering and housekeeping data | |
| | EDL | | |
| | | Pre-EDL | |
| | | Entry | |
| | | Descent | |
| | | Landing | |
| | Perform surface operations | | |
| | | Traverse the Martian surface | |
| | | Acquire and handle samples | |
| | | Evaluate current position via TRS data | |
| | | | Perform reconnaissance activity |
| | | Collect science data | |

2*. Establish the entities to be assessed in Phase Two (RBA).*

a. Although these entities will not be assessed until Phase Two of this process, establishing the relationships between these entities and the system capabilities will provide useful information about the capabilities and about the role of software in meeting those capabilities, which will be useful for completing step #4 of Phase One.
b. These entities should be developer-defined. If not enough information is available to identify developer-defined entities at the time of initial assessment, the IV&V Project can use a reference architecture or an expected set of entities, which could later be updated and re-assessed when developer-defined entities are known.
c. All entities should be identified, including those related to Ground, Mission Operations, COTS, GOTS, etc. It is important not to overlook the existence of these entities. The scoping process may ultimately eliminate them from IV&V work, but they still need to be recognized and assessed to order to have a complete and accurate picture of the overall risk the IV&V Program is helping to mitigate.

d. Example list of entities:

i. – Cruise - GNC

ii. – Cruise - Thermal

iii. – Cruise - Telecom

iv. – Cruise - Power

v. – EDL GNC

vi. – Rover: Startup & Initialization

vii. – Rover: C&DH

viii. – Rover: Remote Sensing Mast

ix. – Rover: SA/SPaH

x. – Rover: Surface Telecom Subsystem

xi. – Rover: Instrument/Payload IF

xii. – Rover: Surface Power Subsystem

xiii. – Rover: Surface Thermal Subsystem

xiv. – Rover: GNC

xv. – System Fault Protection

xvi. – Instruments: MAHLI

xvii. – Instruments: RAD

xviii. – Instruments: ChemMin

xix. – Instruments: ChemCam

xx. – Instruments: SAM

xxi. – Instruments: MARDI

xxii. – Instruments: DAN

xxiii. – Instruments: REMS

xxiv. – Instruments: APXS

xxv. – Instruments: Mastcam

3*. Recognize the existence of relationships between each capability and each entity.*

- If an entity plays a role in achieving a capability, then a relationship is said to exist between the entity and the capability.

- Example (All entities should be mapped.  This example only shows 7 entities for brevity):

| | | | | Cruise - GNC | Cruise - Thermal | Cruise - Telecom | Cruise Power | EDL GNC | Rover: Startup & Initialization | Rover: C&DH |
|---|---|---|---|---|---|---|---|---|---|---|
| Conduct habitability investigations | | | | | | | | | | |
| | Launch to Mars | | | | | | | | | |
| | Cruise to Mars | | | x | x | x | x | | x | x |
| | | Trajectory control | | x | | x | | | | |
| | | Attitude Control | | x | | x | | | | |
| | Approach Mars | | | | | | | x | | |
| | | Trajectory control | | x | | | | x | | |
| | | Attitude Control | | x | | | | | | |
| | Maintain flight systems | | | | | | | | | |
| | | Establish and maintain power | | | | | x | | | x |
| | | Establish and maintain thermal control | | | x | | | | | x |
| | | Perform fault detection | | | | | | | | x |
| | | Establish and maintain communications | | | | x | | | | x |
| | | Gather engineering and housekeeping data | | x | x | x | x | x | x | x |
| | EDL | | | | | | | | | |
| | | Pre-EDL | | | | | | x | | |
| | | Entry | | | | | | x | | |
| | | Descent | | | | | | x | | |
| | | Landing | | | | | | x | | |
| | Perform surface operations | | | | | | | | | |
| | | Traverse the Martian surface | | | | | | | x | x |
| | | Acquire and handle samples | | | | | | | x | x |
| | | Evaluate current position via TRS data | | | | | | | | |
| | | | Perform reconnaissance activity | | | | | | x | x |
| | | Collect science data | | | | | | | x | x |

4*. Establish the role of software for each capability.*

- After the capabilities to be assessed are established, a determination must be made as to whether or not those capabilities will have software associated with them (i.e., whether or not a capability will be implemented by software or affected by software).  The following are some of the things that must be taken into consideration to determine whether a system capability needs to fall into this risk-based assessment:
    - Is the capability expected to be fully automated by software?
    - Is the capability implemented via hardware with software controls?
    - Is the capability decision support or situational awareness related?
    - Is the capability command and control related?
    - Is the capability mission management related (i.e., for planning and executing planned sequences)?
- Once it is understood which capabilities may be associated with software, those capabilities are to be assessed for risk as represented by the risk matrix.  The risk matrix assesses two attributes, Impact and Likelihood, to determine the amount of risk associated with each capability.

5*. Identify the Project Category.*

Project Category is a classification performed by the Agency.  It takes into account cost of mission, political importance, etc.  Go to the website https://nen.nasa.gov/web/pm/ampl and then select “Download a copy of the Agency Mission Directorate Program and Project List (AMPL)”. This will provide you a way to search for your project, which will lead you to the category assigned by the Agency.

6*. For each Capability, assess Impact.*

Impact represents the relative importance of the capability or entity under evaluation. Impact is a measure of the effect of a limitation or issue within the capability under evaluation (Phase One) or of the result of a failure of the entity under evaluation (Phase Two). Generally, you consider the worst case scenario that is **reasonable**.

Impact is based on 3 categories, each scored on a scale from 1 to 5. The Impact Score may also be affected by the Project Category identified above in Step #5. The 3 impact categories are as follows:

- Performance
- Personnel Safety
- Operational Software Control

Criteria for these 3 categories can be found in Appendix A, Assessment Criteria. For each system capability, score each of the 3 categories. Document technical and engineering rationale for each score, clearly explaining how you reached your conclusions and why a particular value was chosen.

Impact Score algorithm: Impact = (max (PS, (AVG (P, OSC) – PCF)))

PS = Personnel Safety

P = Performance

OSC = Operational Software Control

PCF = Project Category Factor:

- Category 1 = 0
- Category 2 = 1
- Category 3 = 2

Impact Score is calculated as follows:

1. Take the average score of Performance and Operational Software Control.
2. If the Project Category is:
    a. Category 1: no change to the result of Step #1.
    b. Category 2: subtract 1 from the result of Step #1.
    c. Category 3: subtract 2 from the result of Step #1.
3. Take the higher of the result from Step #2 and “Personnel Safety”.
4. Round to the nearest Integer. The result of this step is the Impact Score.

7*. For each Capability, assess Likelihood.*

Likelihood is assessed to determine the potential for the existence of errors within the Capability (Phase One) or entity (Phase Two) under evaluation.

Likelihood is based on 4 categories:

- Complexity
- Testability
- Degree of Innovation
- Developer Characteristics

Criteria for these 4 categories can be found in Appendix A, Assessment Criteria. For each system capability, score each of the 4 categories. Document technical and engineering rationale for each score, clearly explaining how you reached your conclusions and why a particular value was chosen.

Likelihood score algorithm: Likelihood = average (complexity, testability, degree of innovation, development characteristics)

Likelihood score is calculated as follows:

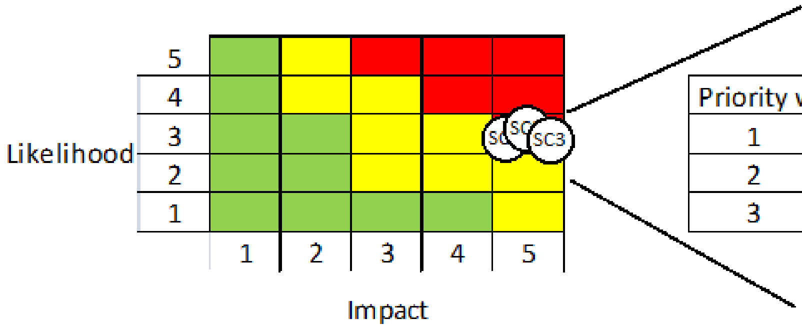
1. Take the average of the scores from the 4 categories.

8*. PBRA Board Review*

Once all of the capabilities have been assessed and the rationale for each has been documented, the results are then provided to the PBRA Board for review. The PBRA Board is chaired by the IV&V Office Lead and includes the IV&V Office Deputy Lead, the Technical Quality and Excellence (TQ&E) Lead, and other members at the discretion of the chair. The PBRA Board is responsible for reviewing and finalizing the scores for each capability. When consensus cannot be reached by the board, the IV&V Program Manager will make the final scoring decision. All final decisions and scores will be communicated to the IV&V Project personnel. This feedback look is intended to ensure that results are understood by all, and to promote consistency in usage of the PBRA in future efforts.

9*. Develop the IV&V Program portfolio*

Once the PBRA Board has completed its review, the IV&V Office Lead may apply a relative importance weight to the scores of each capability. The relative importance weight is derived from the software inventory for which stakeholders at HQ have already provided input. Relative importance must be taken into consideration if all mission capabilities are going to be compared to one another. This enables differentiation among capabilities that result in the same risk score. For example, if three capabilities – SC1, SC2, and SC3 – each have a risk score of 5x3, the relative importance factor will identify which of those three is most important relative to the others. Figure 2 depicts this ranking. Other dimensions need to be taken into consideration as well, such as budget, life cycle state, historical knowledge, etc.




| Priority w/in Matrix Cell | | Risk Score: Impact | Risk Score: Likelihood |
|---|---|---|---|
| 1 | SC2 | 5 | 3 |
| 2 | SC3 | 5 | 3 |
| 3 | SC1 | 5 | 3 |

**Figure 2: Application of relative importance factor integrates the software inventory into the IV&V Portfolio and enables the IV&V Program to prioritize capabilities that have the same risk score.**

In Figure 3, the shading of the matrix demonstrates the relative amount of attention IV&V applies to the capability. Areas of the matrix shaded red indicate the highest level of IV&V attention; areas shaded yellow indicate less IV&V attention; areas shaded green indicate no IV&V attention.

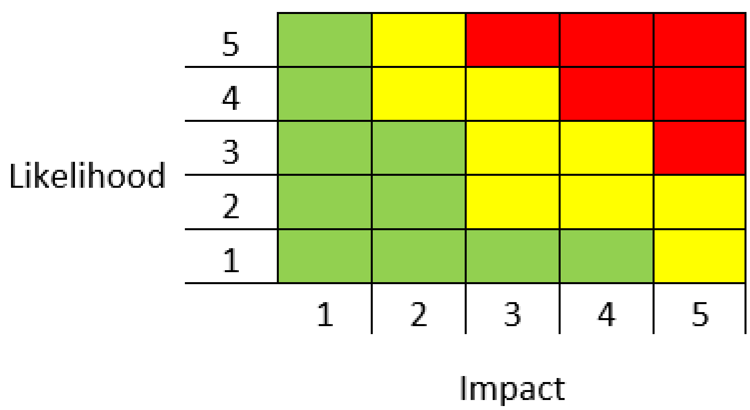



**Figure 3: Risk matrix in which Likelihood is represented by the vertical axis and Impact is represented by the horizontal axis. Red indicates capabilities of high risk that warrant the highest amount of IV&V attention; yellow indicates capabilities of medium risk that warrant lesser IV&V attention; and green indicates low risk that does not warrant any IV&V attention. (NOTE: This shading is reused from the ESMD Risk Management Plan and would need to be updated for purposes of IV&V.)**

As a next step, portfolio dimensions need to be defined, and what it means to have certain risk scores and potential configurations for the portfolio needs to be considered.

For example, a simple configuration for the IV&V Program Portfolio would be:

*Eq. P1: MinMax(Portfolio) = Maximize(Risk(¬PersonnelSafety)) s.t. | Minimize(Cost) U PersonnelSafety*

This says that the IV&V Program wants to maximize the amount of risk mitigated for the Agency (ensuring that all personnel safety risk is mitigated) for the minimum amount of cost. Based on the IV&V Program’s budget, all “what if” scenarios can be run to see what configurations meet the budget because there is a finite set of capabilities.

The recommended dimensions for the IV&V Portfolio are:

- Amount of risk (result of the risk assessment defined above)
- Cost associated with the capability
- Public’s acceptability of the risk
- Time frame in which the risk can be mitigated
- Mission directorate balance

The goal of the ranking algorithm spanning these portfolio dimensions is to maximize the risk coverage for the Agency within the constraints of the overall IV&V budget, while providing an equitable balance across Agency directorates.

<This marks the end of Phase One (PBRA)>

## *Phase Two:  Risk Based Assessment (RBA)*

Figure 4 depicts the RBA Process.

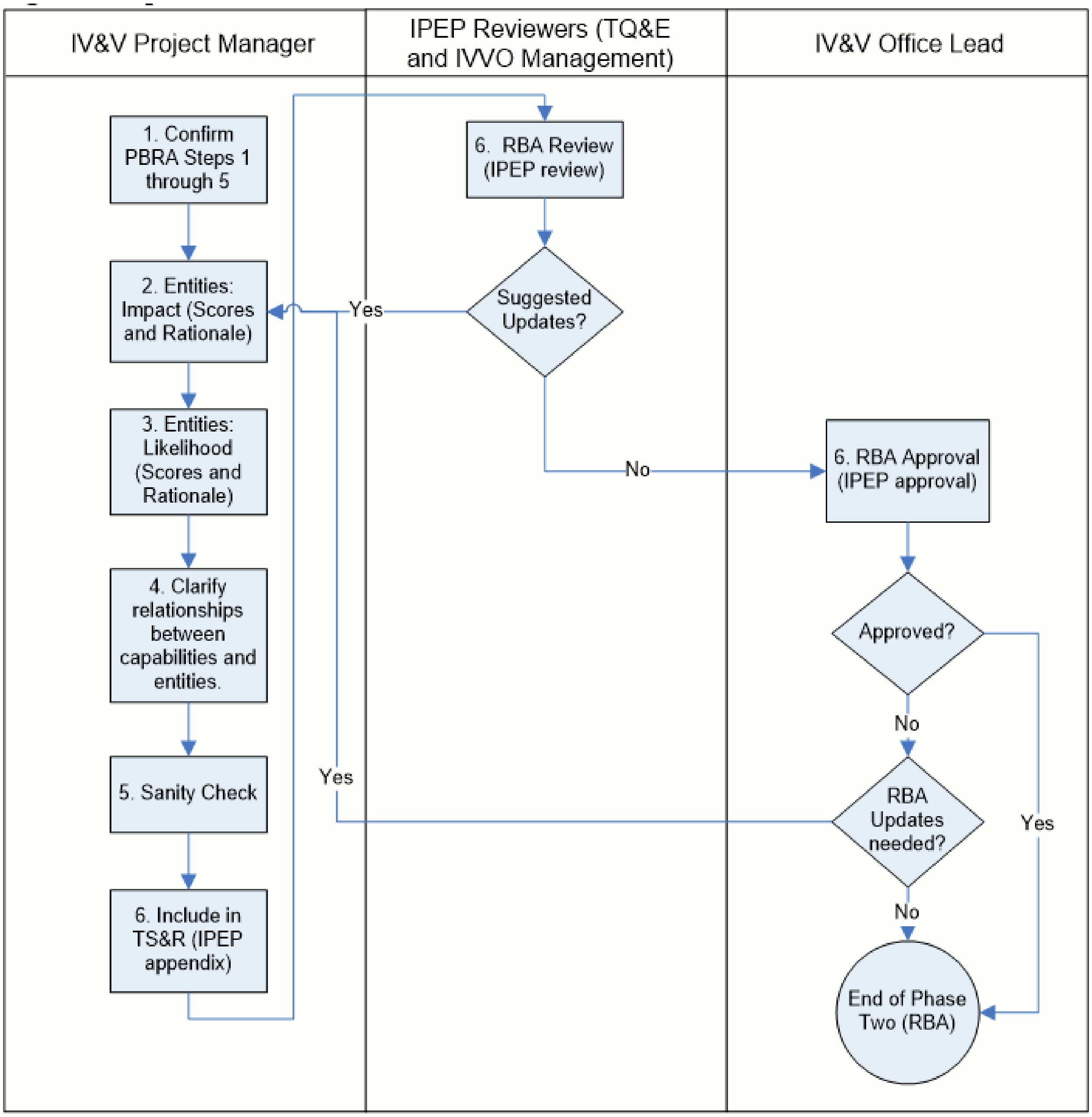



**Figure 4: Risk Based Assessment (RBA) Process**

## Process Steps

1. *Confirm the results of steps 1-5 from Phase One (PBRA)*
    a. Because new information may have become available to IV&V since these steps were executed, it is important to ensure our results are accurate. Repeat steps 1-5 from Phase One (PBRA) as necessary.
    b. If Steps 1-5 from the PBRA process have not yet been executed, they will need to be

executed at this point.

*2. For each entity, assess Impact.*

Impact represents the relative importance of the capability or entity under evaluation. Impact is a measure of the effect of a limitation or issue within the capability under evaluation (Phase One) or of the result of a failure of the entity under evaluation (Phase Two). Generally, you consider the worst case scenario that is reasonable.

Impact is based on 3 categories, each scored on a scale from 1 to 5. The Impact Score may also be affected by the Project Category identified above in Step #5. The 3 impact categories are as follows:

- Performance
- Personnel Safety
- Operational Software Control

Criteria for these 3 categories can be found in Appendix A, Assessment Criteria. For each entity, score each of the 3 categories. Document technical and engineering rationale for each score, clearly explaining how you reached your conclusions and why a particular value was chosen.

Impact Score algorithm: Impact = (max (PS, (AVG (P, OSC) – PCF)))

PS = Personnel Safety

P = Performance

OSC = Operational Software Control

PCF = Project Category Factor:

- Category 1 = 0
- Category 2 = 1
- Category 3 = 2

Impact Score is calculated as follows:

1. Take the average score of Performance and Operational Software Control.
2. If the Project Category is:
    a. Category 1: no change to the result of Step #1.
    b. Category 2: subtract 1 from the result of Step #1.
    c. Category 3: subtract 2 from the result of Step #1.
3. Take the higher of the result from Step #2" and "Personnel Safety".
4. Round to the nearest Integer. The result of this step is the Impact Score.

*3. For each Entity, assess Likelihood.*

Likelihood is assessed to determine the potential for the existence of errors within the Capability (Phase One) or entity (Phase Two) under evaluation.

Likelihood is based on 4 categories:

- Complexity
- Testability
- Degree of Innovation
- Developer Characteristics

Criteria for these 4 categories can be found in Appendix A, Assessment Criteria. For each entity, score each of the 4 categories. Document technical and engineering rationale for each score, clearly explaining how you reached your conclusions and why a particular value was chosen.

Likelihood score algorithm: Likelihood = average (complexity, testability, degree of innovation, development characteristics)

Likelihood score is calculated as follows:

1. Take the average of the scores from the 4 categories.

*4. Clarify the relationships between each capability and each entity*

The RBA process described in this document is used for planning and scoping a NASA IV&V Project. The updated entity-to-capability mapping produced by this step (example below in 4.b) is intended to be a view of the system that serves as a useful tool for discussing and deciding where to apply IV&V effort.

a. For each entity, indicate the area or areas (e.g. capabilities/behaviors) that were the driver for the score for that entity by marking that relationship with “XX”. The rationale for the entity’s scoring should explicitly or implicitly refer to this area or areas (e.g. capabilities/behaviors). To help identify driving relationships, ask, “What is the most important thing this entity does?”
    i. For example, if Cruise - Power is scored 3-1, and the reason it is scored 3-1 is due to its role in “Establish and maintain power”, then that relationship should be marked with “XX”. Similarly, if Rover: C&DH is scored 5-1, and the reason it is scored 5-1 is due to its role in both “Gather engineering and housekeeping data” and its role in “Collect science data”, then both those relationships should be marked with “XX”.
b. Example (with entity scoring and updated entity-to-capability mapping):

| Capability | Cruise - GNC | Cruise - Thermal | Cruise - Telecom | Cruise Power | EDL GNC | Rover: Startup & Initialization | Rover: C&DH |
|---|---|---|---|---|---|---|---|
| Entity Score (Impact - Likelihood): | 5 - 1 | 2 - 1 | 2 - 1 | 3 - 1 | 5 - 2 | 4 - 1 | 5 - 1 |
| Conduct habitability investigations | | | | | | | |
| Launch to Mars | | | | | | | |
| Cruise to Mars | x | x | x | x | | x | x |
| Trajectory control | **xx** | | x | | | | |
| Attitude Control | **xx** | | x | | | | |
| Approach Mars | | | | | **xx** | | |
| Trajectory control | **xx** | | | | **xx** | | |
| Attitude Control | **xx** | | | | | | |
| Maintain flight systems | | | | | | | |
| Establish and maintain power | | | | **xx** | | | x |
| Establish and maintain thermal control | | **xx** | | | | | x |
| Perform fault detection | | | | | | | x |
| Establish and maintain communications | | | **xx** | | | | x |
| Gather engineering and housekeeping data | x | x | x | x | x | x | **xx** |
| EDL | | | | | | | |
| Pre-EDL | | | | | **xx** | | |
| Entry | | | | | **xx** | | |
| Descent | | | | | **xx** | | |
| Landing | | | | | **xx** | | |
| Perform surface operations | | | | | | | |
| Traverse the Martian surface | | | | | | x | x |
| Acquire and handle samples | | | | | | x | x |
| Evaluate current position via TRS data | | | | | | | |
| Perform reconnaissance activity | | | | | | x | x |
| Collect science data | | | | | | x | **xx** |

*5. Perform sanity check*

a. Now that Capabilities and entities have both been scored and relationships have been established and clarified, take the opportunity to evaluate the scoring and rationale to make sure everything seems reasonable.

*6. Include Scoping information in the Technical Scope and Rigor (TS&R) document (IPEP appendix)*

a. IPEP review and approval serves as the feedback and approval mechanism for RBA results. IVV 09-4 Project Management is the authority on IPEP review and approval. Current reviewers are the TQ&E Group and IV&V Office Management. Current approver IV&V Office Lead.

## *Appendix A (Assessment criteria)*

Some general notes regarding the assessment criteria found in this appendix:

- The intent is not to use the criteria as extremely rigid requirements; instead, the criteria are starting points. The intent is to consistently provide thorough, reasonable, and well-documented scores and scoring rationale.
- Two main factors are assessed: Impact and Likelihood
  - Impact criteria are below on a single page
  - Likelihood criteria are below, spread across three pages

- Several of the categories within Impact and Likelihood have “elaborated criteria”. The basic criteria come almost entirely from the original PBRA process, released in December of 2008, and are often high level. “Elaborated criteria” (along with the RBA processes) were produced by an assessment team in March 2010, and serve as additional content that evaluators may find helpful when assessing lower level entities.

| Impact | Very Low (1) | Low (2) | Moderate (3) | High (4) | Very High (5) |
|---|---|---|---|---|---|
| **Performance** | Minimal or No Mission Impact | Minor Impact to Full Mission | Moderate Impact to Full Mission | Major Impact to Full Mission | Loss of Minimum Mission Objectives |
| Elaborated Criteria | • Failure could cause an inconvenience but no impact to mission success, science value, or cost of operation. | • Reduced system performance that does not result in the loss of a mission objective.<br>• Reductions could include short term loss of science collection, implementation of workarounds with minimal cost, or noticeable but minor impact to science value. | • Loss of a single mission objective (mission success or mission return) or degradation in operational performance.<br>• Minimum mission success criteria are met.<br>• Performance degradation may result in costly recovery options, reduced science value, or long term delays to the accomplishment of science. | • Loss of multiple mission objectives<br>• Some science value is retained<br>• Damage to other spacecraft/assets | • Permanent loss of all mission objectives.<br><br>Examples:<br>- Loss of spacecraft<br>- Loss of other spacecraft/assets<br>- Loss of ability to collect science data<br>- Loss of primary instrument |
| **Personnel Safety**[1] | No Injury | Minor Injury/Illness (ref. 8621.1B Type D) | Lost Time Injury/Illness (ref. 8621.1B Type C) | Permanent Partial Disability (ref. 8621.1B Type B) | Death, Permanent Total Disability (ref. 8621.1B Type A) |
| Elaborated Criteria | NA | NA | NA | NA | NA |
| **Operational Software Control**[2] | Software does not control safety-critical hardware systems, subsystems or components and does not provide safety-critical information. | Software does not control safety-critical hardware systems, subsystems or components and does not provide safety-critical information. However, software resides within a computing device such that failure of the device has the potential for a Level 3 performance impact. | Software item issues commands over potentially hazardous hardware systems, subsystems or components requiring human action to complete the control function. There are several, redundant, independent safety measures for each hazardous event.<br><br>Software generates information of a safety-critical nature used to make safety-critical decisions. There are several redundant, independent safety measures for each hazardous event.<br><br>Software does not control safety-critical hardware systems, subsystems or components and does not provide safety-critical information. However, software resides within a computing device such that failure of the device has the potential for a Level 4 or 5 performance impact. | Software exercises control over potentially hazardous hardware systems, subsystems, or components allowing time for intervention by independent safety systems to mitigate the hazard. However, these systems by themselves are not considered adequate.<br><br>Software item displays information requiring immediate operator action to mitigate a hazard. Software failures will allow, or fail to prevent, the hazard's occurrence. | Software exercises autonomous control over potentially hazardous hardware systems, subsystems or components without the possibility of intervention to preclude the occurrence of a hazard. Failure of the software, or a failure to prevent an event, leads directly to a hazard's occurrence. |
| Elaborated Criteria | NA | NA | NA | NA | NA |

[1] 8621.1B effective date May 23, 2006 Chapter 1, Figure 1

[2] "Operational Software Control" is based almost entirely on the "Software Control Categories" found in NASA Software Safety Guidebook (NASA-GB-8719.13), Table 3-1 MIL STD 882C Software Control Categories. Content was modified to shift from a 4 point scale to a 5 point scale, and to account for software that resides within a computing device such that failure of the device will lead to a certain level performance impact.

| Likelihood | Very Low (1) | Low (2) | Moderate (3) | High (4) | Very High (5) |
|---|---|---|---|---|---|
| **Complexity** | Simple and straightforward, nothing distributed and limited interfaces. | Relatively basic capability with states and transitions internal to the element. Element capability is realized as a result of combining the behaviors and internal interface information. | Relatively basic capability but states and transitions between elements effect the behavior. Capability is realized as a result of combining the behaviors of several objects with the same or similar interfaces. | Moderately complex, broad engineering community understanding and capability is realized as a result of combining the behaviors across several internal and external interfaces. | Very complex, few understand the capability and capability is realized as a result of combining the behaviors of several objects with different interfaces. |
| Elaborated Criteria | Straight-line code with few to no nested structured programming operators: DOs, CASEs, IF THEN ELSEs. Simple module composition via procedure calls or simple scripts.<br><br>Simple read-write statements with simple formats. Simple COTS-DB queries and updates.<br><br>Function operates in only one more of system operation.<br><br>Evaluation of simple expressions. | | Simple nesting with some inter-module control including decision tables, message passing and middleware supported distributed processing. Simple I/O processing including status checking and error processing.<br><br>Multi-file input or single file input with minimal structural changes to the files.<br><br>Function behaves differently in different modes of system operation.<br><br>Standard math and statistical routines to include basic vector operations. | | Multiple resource scheduling with dynamically changing priorities or distributed real-time control.<br><br>Performance critical embedded system. Highly coupled dynamic relational and object structures.<br><br>Object uses different end items (sensors) in different modes (stages) of system operation.<br><br>Difficult and unstructured numerical analysis: highly accurate analysis of noisy, stochastic data and/or complex parallelization. |

| Likelihood | Very Low (1) | Low (2) | Moderate (3) | High (4) | Very High (5) |
|---|---|---|---|---|---|
| **Testability** | Simple path to be exercised, input required to stimulate execution path is easily identified and finite, and output easily logged can be automatically compared to success criteria. | Complex path to be exercised, input required to stimulate execution path is identified but large, and output is compared to success criteria automatically. | One or more paths required to exercise the capability, input required to stimulate execution path may be infinite but easily classified (e.g. equivalence classes), some input dependent on emulators and simulators but not all. Assessing output is fairly straightforward (e.g. some results may require analysis). | Multiple paths required to exercise the capability, input required to stimulate execution path may be infinite with a few difficult concepts, input is also partially dependent on emulators and simulators. Assessing output is partially dependent on analysis. | Multiple paths required to exercise the capability, input required to stimulate execution path may be infinite or difficult to conceptualize, input is also entirely dependent on emulators and simulators. Assessing output is entirely dependent on analysis. |
| Elaborated Criteria | A scriptable interface or test harness is available.<br>Software and hardware states and variables can be controlled directly by the test engineer.<br>Software modules, objects, or functional layers can be tested independently (low level of coupling).<br>Test expectations are fully quantified.<br>Past system states and variables are visible or queriable (e.g., transaction logs).<br>Current system states and variables are visible or queriable during the execution.<br>Distinct output is generated for each input.<br>System states and variables are visible or queriable during execution.<br>All factors affecting the output are visible.<br>Incorrect output is easily identified.<br>Internal errors are automatically detected and reported through self-testing mechanisms.<br>Module can be fully tested via inspection. | Tests are written before coding is performed.<br>Testing is not wholly independent, but only 1 or 2 other interfaces are required.<br>The majority of system states and variables are visible or queriable during execution.<br>Internal errors are automatically detected but requirement manual correction (no self-testing mechanism). | Software and hardware states can be influenced or indirectly controlled by the test engineer.<br>Not all factors affecting the output are visible.<br>Module is not singular in responsibility, i.e., mid-level cohesiveness.<br>Determination of the correctness of the output may require some limited analysis.<br>Test expectations are available, but may not be fully documented.<br>Testing of the module is dependent on a limited number of other modules (mid-level coupling). | Partial visibility of past system states and variables.<br>Partial insight into the current state of the module/system component during testing.<br>Testing through demonstration is acceptable.<br>Some test expectations are non-quantifiable.<br>Testing is reliant on multiple interfaces, many simulated in order to execute the software. | Testing is not considered until coding is complete.<br>Software and hardware states cannot be directly controlled by the test engineer.<br>Software module cannot be independently tested (high level of coupling) without multiple simulated interfaces.<br>Past system states and variables are not visible.<br>Generated output cannot be directly derived from the provided input.<br>Incorrect output is not easily identified - requires manual analysis.<br>Low cohesiveness.<br>Test expectations are unknown or non-quantifiable. |

| Likelihood | Very Low (1) | Low (2) | Moderate (3) | High (4) | Very High (5) |
|---|---|---|---|---|---|
| **Degree of innovation** | Capability has been developed before by this team and has flown on several missions | Capability has flown several missions, but has been developed by another team | Capability has flown before, fairly mature and well know, but is being modified for mission | Capability has flown only one mission, but is modified based on data from that mission | Capability is being proven on mission and limited experience in developing like-capability |
| Elaborated Criteria | • Proven on other systems with same application<br>• Mature experience<br>• Well documented testing<br>• Solid requirements - little potential for change<br>• Little to no integration required<br>• No interaction with multiple organizations<br>• Actual system "Flight Proven" through successful mission operations. | System prototype demonstration in a space environment -or- actual system complete and "Flight Qualified" through test and demonstration (ground or space). | System/subsystem model or prototype demonstration in a relevant environment. | Component and/or breadboard validation in a laboratory environment -or- Component and/or breadboard validation in relevant environment. | Basic principles observed and reported -or- Technology concept and/or application formulated -or- Analytical & Experimental critical functions and/or characteristics proof-of-concept. |
| **Development Characteristics** | Developer uses a mature engineering approach and makes use of a documented and tried process (industry wide or local) | Developer uses new engineering approaches which are documented and followed | Developer has a mature process planned and evidence suggest that the planned processes are not being followed | Developer has a mature engineering process planned but actual implementation of the process is incomplete and ad hoc engineering is completing them | Developer's engineering approach is ad hoc with minimal documentation as well as planning |
| Elaborated Criteria | Developed more than one like system or current incumbent<br><br>Developer does not use subcontractors and developer staff/management are co-located | Developed one like system<br><br>Developer does use subcontractor(s) and developer staff/management are co-located | Nominal domain or related experience (10+ years)<br><br>Developer does not use subcontractor and developer staff/management are not co-located | Some domain or related experience (5-10 years)<br><br>Developer uses one subcontractor and management/staff that are not co-located (i.e., geographically dispersed) | Minimal domain or related experience (less than 5 years)<br><br>Developer uses multiple subcontractors and management/staff that are not co-located (i.e., geographically dispersed) |

---

[1] NASA Software Safety Standard (NASA-STD-8719.13B)

## References

| REFERENCES | |
|---|---|
| **Document ID/Link** | **Title** |
| IVV QM | NASA IV&V Quality Manual |
| IVV 09-4 | Project Management |
| NASA-GB-8719.13 | NASA Software Safety Guidebook |
| NASA-STD-8719.13B | NASA Software Safety Standard |
| NPR 8621.1 | NASA Procedural Requirements for Mishap and Close Call Reporting, Investigating, and Recordkeeping |
| | |

**If any procedure, method, or step in this document conflicts with any document in the NASA Online Directives Information System (NODIS), this document shall be superseded by the NODIS document. Any external reference shall be monitored by the Document Owner for current versioning.**

## Version History

| VERSION HISTORY | | | |
|---|---|---|---|
| **Version** | **Description of Change** | **Author** | **Effective Date** |
| Basic | Initial Release | Christina Moats | 12/11/2008 |
| A | Made minor revisions for clarity | Christina Moats | 1/13/2009 |
| B | Major revision, combining the PBRA process with the RBA process produced by an assessment team in March 2010 | Jeff Northey | 2/23/2011 |
| C | Remove ambiguous use of IV&V Coverage and coverage categories | Patrick Theeke, et. al. | 4/17/2012 |

| | | | |
|---|---|---|---|
| D | Updated based on PAR 2012-P-364: Added Safety Criteria: Damage/loss to other spacecraft/assets. Fixed URL. Reworded some Complexity. | Steve Husty | 10/4/2012 |
| | | | |