System Risk Balancing Profiles: Software Component

John C. Kelly & Burton C. Sigal
Quality Assurance Office

Tom Gindorf
Assurance Technology Program Office

NASA Jet Propulsion Laboratory,
Pasadena, CA

Office of Engineering and Mission Assurance, Assurance Technology Programs
Background*

- NASA's new environment:
  - From Few to Many Projects
  - From Large to Small Projects
  - From Single Monumental Success to Many Opportunities for Success
  - From a Large Budgets to Declining or Flat Budgets
  - From Conservative Risk Avoidance to Risk Management based on cost of failure

- Risk is a resource that can be traded like other resources (mass, power, performance, schedule, & cost)

- For the reasonably small number of critical subsystems which could affect human life, the traditional risk avoidance approach should still be used

*Based on Greenfield & Gindorf, “Risk as a Resource - A New Paradigm”, 1996
What can be done to balance the wide variety of risks a project needs to manage?
Approach

- Provide a mechanism for identifying performance risk associated with program content
- Identify mitigation possibilities corresponding to residual performance risk
- The full set of charts will address balancing risk involving System, Hardware and Software
- Risk associated with people, organizations, and facilities are not within the scope of this work, but need to be addressed separately
Overview:
“FODORs Charts”

High Risk

Resources

Program Content

Residual Risks

Mitigations

Low Risk
Resources*

- Drivers
  - Spacecraft & Science Performance
  - Cost
  - Schedule
- Planning Cautions
  - Cost Risk Factors
  - Schedule Risk Factors
- Mass
- Power

* Note: Resources include some areas which may not be applicable to software (i.e. mass & power)
Software Quality Assurance / V&V Program Content

- **Component Areas**
  - Testing
  - Analysis
  - Quality Assurance
  - Related Management
  - Other

- **Used common “tried and proven” software QA / V&V activities to populate the program content**

- **Used a history of QA / V&V services that were previously provided for JPL and NASA projects (on project funds)**

- **Advanced approaches, software QA / V&V research, and “piated only” techniques were not included in the program content area**

- **Content ranged from a super minimal approach to a full up QA / V&V program**
Software Quality Assurance
/ V&V Program Content (continued)

- Included software safety and hazards analysis as required by NASA policy even in the minimal program category
- SEI's Capability Maturity Model was originally used to segment Software Quality and V&V into consistent levels
- Qualitative differences in the individual program content activities were noted
  - Acceptance Test (pass/fail)
  - Acceptance Test (w/metrics & key critical functions)
  - Acceptance Test (w/metrics, good functional coverage, & witnessing)
  - Acceptance Test (w/metrics, full functional coverage, & witnessing)
Residual Performance Risks

• “What are the risks, if projects chose not to do individual program content items”?
  – We went through the program content list asking ourselves:
    • “If this QA /V&V activity is deleted, what can/has go wrong? and
    • “If this QA /V&V activity is used correctly, what problems/risks should be avoidable?

• In today’s NASA environment, the full up “Low Risk” QA / V&V program can only be justified for a few isolated projects

• Raised excellent questions regarding the content area from a project management viewpoint:
  • If I don’t choose or have funds to have particular QA / V&V program content areas, what risks are being accepted by the project?
  • Are there redundancies in program content items with respect to individual risks?
  • Are there risks that have insufficient coverage by standard QA / V&V program content areas?
  • Given a limited budget and specific project resource drivers for QA / V&V, is the project buying the best program content?
Mitigations

- Project factors or techniques which reduce/eliminate the risks associated with software
- Given a set of typical program content items, risks can be reduced by:
  - Adding program content from another column when considering the aggregate of risks
  - Utilizing mitigations from the same column
- Mitigations include
  - Advanced techniques (Formal Methods, Model Checking, Simulation, etc.)
  - Opportunistic factors (reusing high quality software components, etc.)
- Mitigations need to be carefully selected with assistance from someone with expertise in a broad spectrum of software QA / V&V techniques
### The Early Mapping of SEI's Capability Maturity Model into the FODOR Chart

<table>
<thead>
<tr>
<th>SEI Level 1 KPAs</th>
<th>SEI Level 2 KPAs</th>
<th>Tailored Approach</th>
<th>SEI Level 2 &amp; 3 KPAs</th>
<th>SEI Level 2, 3, 4 &amp; 5 KPAs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration Management (K1)</td>
<td>Software Quality Assurance (K2)</td>
<td>S/W Subcontract Management (K3)</td>
<td>S/W Project Tracking and Oversight (K4)</td>
<td>S/W Project Planning (K5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Requirements Management (K6)</td>
<td>Peer Review (K7)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Intergroup Coordination (K8)</td>
<td>S/W Product Engineering (K9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S/W Product Engineering (K9)</td>
<td>Integrated S/W Management (K10)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Training Program (K11)</td>
<td>Org. Process Definition (K12)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Org. Process Focus (K13)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration Management (K1)</td>
<td>Software Quality Assurance (K2)</td>
<td>S/W Subcontract Management (K3)</td>
<td>S/W Project Tracking and Oversight (K4)</td>
<td>S/W Project Planning (K5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Requirements Management (K6)</td>
<td>Peer Review (K7)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Intergroup Coordination (K8)</td>
<td>S/W Product Engineering (K9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S/W Product Engineering (K9)</td>
<td>Integrated S/W Management (K10)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Training Program (K11)</td>
<td>Org. Process Definition (K12)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Org. Process Focus (K13)</td>
<td></td>
</tr>
</tbody>
</table>
Early version of Guide

- <insert version 20 of the guide here, see attached file “A_Balance.doc”>
Evolution of the Guide

- Reduced the number of columns from 5 to 3 to show only the extremes
- Introduced identifiers to trace
  - Missing or Weak Program Content to Residual Risks
  - Mitigations to Residual Risk
Software QA / V&V Guide

- <insert the latest guide here, see the attached file “B_Balance.doc”>
“Reading” the Guide

- The guide is a starting place to tailor a project/mission in a number of support domains. The center column is for the "results" of the tailoring decisions and could be a basis for the Software QA or V&V Plan.
- The left columns represent a High Risk (low Software QA / V&V content) while the right hand column represents a Low Risk (high Software QA / V&V content) approach.
- The top "group" of each column (Program) contains the software related activities, divided into five (5) areas, Testing, Analysis, QA, (Related) Management, and Other. Each element has a reference designator for tractability purposes.
- The center group (Residual Performance Risks) contains the residual risks that occur because of the activities that are NOT to be done by the program.
“Reading” the Guide (continued)

- After identifying the residual risks, there are choices. The programs can:
  - Use some/all of the mitigation strategies in the lower group (Mitigations) to reduce/mitigate the associated risk,
  - Change their minds and do the (upper group) activity to eliminate risks,
  - Do both,
  - Decide to do nothing and accept the risk.

- The numbers after each mitigation strategy trace to the residual risk it is intended to mitigate. Notice that several strategies address the same risk, and many strategies address multiple risks, so the projects will have a number of cost-benefit tradeoffs they can make in managing risk.

- There are no 100% certain, 0% Risk programs!
What the Guide “is” and what it “is not”

- The Guide is:
  - Useful for identifying project risk associated with a level of QA /V&V program content
  - Identifying mitigation possibilities
  - Helpful in planning appropriate resources for QA / V&V program content (and balancing resources across various project risk reduction areas)

- The Guide is not:
  - a substitute for an experts’ participation during the planning process
  - prescriptive in nature (it is intended to illustrate how to tailor a QA / V&V program)
  - a process monitoring and corrective action technique (needed by projects beyond the use of this guide)
Summary

- The Software QA / V&V guide will be reviewed and updated based on feedback from NASA organizations and others with a vested interest in this area.
- Hardware, EEE Parts, Reliability, and Systems Safety are a sample of the future guides that will be developed.
- Cost Estimates, Lessons Learned, Probability of Failure and PACTS (Prevention, Avoidance, Control or Test) are needed to provide a more complete risk management strategy.
- This approach to risk management is designed to help balance the resources and program content for risk reduction for NASA’s changing environment.
## Risk Balance Profile

**Software Quality and V&V Program Guide**

### FODORS

<table>
<thead>
<tr>
<th>Program Title</th>
<th>Very High Risk</th>
<th>Medium/High Risk</th>
<th>Tailored Approach</th>
<th>Medium Risk</th>
<th>Complete QA/V&amp;V Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1- Acceptance Test (pass/fail)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2- Functional Test (pass/fail)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1- Hazards Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2- S/W FMEA of critical functions only</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QA</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Related Management</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### RESIDUAL RISKS

<table>
<thead>
<tr>
<th>RESIDUAL RISKS</th>
<th>Q1, Q3</th>
<th>Q2, Q5</th>
<th>Q6, Q7</th>
<th>Q8, Q9</th>
<th>Q10, Q11, Q12, Q13, Q14, Q15, Q16</th>
<th>Q17, Q18, Q19, Q20, Q21, Q22, Q23, Q24, Q25, Q26</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1 - Lack of confidence in acceptability of S/W to meet system's needs-T1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2 - Unknown functional and system margins-T2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R3 - Inconsistent S/W requirements with respect to the system's functional requirements (FRD/Q2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R4 - Incorrect design functionality-Q2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P2 - S/W builds not converging to an acceptable product - T5, M2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R7 - Inputs to S/W could violate boundary conditions, trigger non-tested paths, etc. - T5, Q2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R8 - Poor Workmanship in the software product (spaghetti code, un-maintainable code, etc.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R13 - Lack of robustness of functions supported by S/W - Q9, Q5, Q4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R14 - S/W fails in a harmful manner - A1, A2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R15 - Latent S/W defects could cause the system to fail or not meet it's requirements - T5, Q2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R16 - Latent S/W defects could cause the system to fail or not meet it's requirements - T5, Q2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R17 - Late awareness (or lack of anticipation) of schedule performance, cost and quality problems - T5, Q5, M2, M3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R18 - Software safety problem - A2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R19 - Executing faulty commands on a spacecraft - Q1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R20 - Lack of robustness of functions supported by S/W - Q9, Q5, A4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R21 - S/W fails in a harmful manner - A1, A2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R22 - Late awareness (or lack of anticipation) of schedule performance, cost and quality problems - T5, Q5, M2, M3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R23 - Software safety problem - A2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R24 - Executing faulty commands on a spacecraft - Q1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SOFTWARE RISK FACTORS

- Schedule pressure resolved by $ (Schedule Pressure Resolved by $)
- Repeat Testing
- Changing Requirements
- S/W Faults Causing Impact

### PROGRAM SCHEDULE RISK FACTORS

- Late Problem Identification
- Repair and Repeat Testing
- Changing Requirements
- S/W Faults Causing Impact

### GENERAL ELEMENTS

- Negligible or small
- Negligible or small
- Negligible or small
- Negligible or small

### Tailored Approach

- As Selected (Tailored to be Project Specific)

### Related Management

- MI- Full Q/A Plan
- MI- Configuration Management (Code & version control)
- MI- Milestone Reviews (CDR, PDR, DCR)
- MI- Risk Management program (basic)
- MI- Project S/W Metrics program (System/Acc. P/F/Ra)
- MI- Support Contractor Mgt. (Assessment of critical areas)

### Software QA & V&V (incremental updates)

### Related Management

- MI- Full S/W QA Plan
- MI- Configuration Management (Code & Version Control)
- MI- Milestone Reviews (CDR, PDR, DCR)
- MI- Risk Management program (basic)
- MI- Project S/W Metrics program (System/Acc. P/F/Ra)

### Residual Risks

- R1 - Lack of confidence in acceptability of S/W to meet system's needs-T1
- R7 - Inputs to S/W could violate boundary conditions, trigger non-tested paths, etc. - T5, Q2
- R8 - Poor Workmanship in the software product (spaghetti code, un-maintainable code, etc.) - Q1, Q3
- R9 - Latent S/W defects could cause the system to fail or not meet it's requirements - T5, Q2, Q5
- R10 - Late awareness (or lack of anticipation) of schedule performance, cost and quality problems - T5, Q5, M2, M3
- R11 - Software safety problem - A2, A3
- R12 - Executing faulty commands on a spacecraft - Q1, Q2
- R13 - Lack of robustness of functions supported by S/W - Q9, Q5, A4
- R14 - S/W fails in a harmful manner - A1, A2

### Relevant Risks

- R7 - Inputs to S/W could violate boundary conditions, trigger non-tested paths, etc. - T5, Q2
- R9 - Latent S/W defects could cause the system to fail or not meet it's requirements - T5, Q2, Q5
- R11 - Software safety problem - A2, A3
- R12 - Executing faulty commands on a spacecraft - Q1, Q2
- R13 - Lack of robustness of functions supported by S/W - Q9, Q5, A4
- R14 - S/W fails in a harmful manner - A1, A2

### Risk Protection

- MI- Full S/W QA Plan
- MI- Configuration Management (Coding, testing, Version control)
- MI- Milestone Reviews (CDR, PDR, etc.)
- MI- Risk Management program (basic)
- MI- Project S/W Metrics program (System/Acc. P/F/Ra)

### Related Management

- MI- Support Contractor Mgt. (Assessment of critical areas)
- MI- Mission Operations and Command Assurance (MOCA)
- MI- Milestone Reviews (CDR, PDR, etc.) with participation of independent reviewers mandatory
- MI- Milestone Reviews (CDR, PDR, etc.) with participation of independent reviewers mandatory
- MI- Project Safety Management program
- MI- Integrated Support of Fault Protection and/or Failure Detection, Isolation & Recovery subsystems
- MI- Full Project Software Metrics program

### Other

- MI- Support Contractor Mgt. (Assessment of critical areas)
- MI- Mission Operations and Command Assurance (MOCA)
- MI- Milestone Reviews (CDR, PDR, etc.) with participation of independent reviewers mandatory
- MI- Project Safety Management program
- MI- Integrated Support of Fault Protection and/or Failure Detection, Isolation & Recovery subsystems
- MI- Full Project Software Metrics program

### Appendix

- MI- Support Contractor Mgt. (Assessment of critical areas)
- MI- Mission Operations and Command Assurance (MOCA)
- MI- Milestone Reviews (CDR, PDR, etc.) with participation of independent reviewers mandatory
- MI- Project Safety Management program
- MI- Integrated Support of Fault Protection and/or Failure Detection, Isolation & Recovery subsystems
- MI- Full Project Software Metrics program

### 10/21/98

V19FODORS.DOC
<table>
<thead>
<tr>
<th>Mitigations</th>
<th>Mitigations</th>
<th>Mitigations</th>
<th>Mitigations</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S1) Use of an Automatic Code Generator</td>
<td>(S1) Use of an Automatic Code Generator</td>
<td>(S2) Reusing high quality proven software products (req., design, code, and/or test cases)</td>
<td>(S2) Reusing high quality proven software products (req., design, code, and/or test cases)</td>
</tr>
<tr>
<td>(S2) Validation of auto code generator</td>
<td>(S2) Validation of auto code generator</td>
<td>(S3) Rapid Prototyping aspects of the software system</td>
<td>(S3) Rapid Prototyping aspects of the software system</td>
</tr>
<tr>
<td>(S3) Establish reuse requirements</td>
<td>(S3) Establish reuse requirements</td>
<td>(S4) Simulation of software subsystem</td>
<td>(S4) Simulation of software subsystem</td>
</tr>
<tr>
<td>(S4) Reusing high quality proven software products (req., design, code, and/or test cases)</td>
<td>(S4) Reusing high quality proven software products (req., design, code, and/or test cases)</td>
<td>(S5) Embedding Assertions in the code for more thorough testing and insight into the operation of the software (i.e., instrumenting to code)</td>
<td>(S5) Embedding Assertions in the code for more thorough testing and insight into the operation of the software (i.e., instrumenting to code)</td>
</tr>
<tr>
<td>(S5) Early training</td>
<td>(S5) Early training</td>
<td>(S6) Lessons learned</td>
<td>(S6) Lessons learned</td>
</tr>
<tr>
<td>(S6) Incentivize contractor</td>
<td>(S6) Incentivize contractor</td>
<td>(S7) Apply PACTS to critical functions</td>
<td>(S7) Apply PACTS to critical functions</td>
</tr>
<tr>
<td>(S7) Do regression testing</td>
<td>(S7) Do regression testing</td>
<td>(S8) Analyze for critical functions</td>
<td>(S8) Analyze for critical functions</td>
</tr>
<tr>
<td>(S8) Cross training</td>
<td>(S8) Cross training</td>
<td>(S9) Establish volatility metrics</td>
<td>(S9) Establish volatility metrics</td>
</tr>
<tr>
<td>(S9) Do regression testing</td>
<td>(S9) Do regression testing</td>
<td>(S10) Early training</td>
<td>(S10) Early training</td>
</tr>
<tr>
<td>(S10) Cross training</td>
<td>(S10) Cross training</td>
<td>(S11) Use Automation techniques with Formal Methods techniques (formal specification, model checking, animating specifications, and/or proofs))</td>
<td>(S11) Use Automation techniques with Formal Methods techniques (formal specification, model checking, animating specifications, and/or proofs))</td>
</tr>
<tr>
<td>(S11) Early training</td>
<td>(S11) Early training</td>
<td>(S12) Use Automation techniques with Formal Methods techniques (formal specification, model checking, animating specifications, and/or proofs))</td>
<td>(S12) Use Automation techniques with Formal Methods techniques (formal specification, model checking, animating specifications, and/or proofs))</td>
</tr>
<tr>
<td>(S12) Cross training</td>
<td>(S12) Cross training</td>
<td>(S13) Incentivize contractor</td>
<td>(S13) Incentivize contractor</td>
</tr>
<tr>
<td>(S13) Use TSAs (incl auto code gen)</td>
<td>(S13) Use TSAs (incl auto code gen)</td>
<td>(S14) Use TSAs (incl auto code gen)</td>
<td>(S14) Use TSAs (incl auto code gen)</td>
</tr>
</tbody>
</table>
## Risk Balance Profile

**Software Quality and V&V Program Guide**

**“FODORS”**

<table>
<thead>
<tr>
<th>Program Content</th>
<th>Minimal QA/V&amp;V Program (Very High Risk)</th>
<th>Tailored Approach</th>
<th>Complete QA/V&amp;V Program (Low Risk)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Testing</strong></td>
<td>T1-Accept Test (pass/fail w/o metrics)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>T2-Functional Test (pass/fail)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Analysis</strong></td>
<td>A1-Hazards Analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A2/S/W FMEA (if applicable for critical functions only)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>QA</strong></td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Related Management</strong></td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Residual Risks (Missing Content)**

1. Lack of confidence in acceptability of S/W to meet system’s needs - T1+
2. Unknown functional and system margins - T2+
3. Inconsistent S/W requirements with respect to the system’s functional requirements (FRD) - Q2
4. Incorrect design functionality - Q2
5. No regression testing - T5, M4
6. S/W builds not converging to an acceptable product - T5, M2
7. Inputs to S/W could violate boundary conditions, trigger non-tested paths, etc. - T5, Q2
8. Poor Workmanship in the software product (spaghetti code, unmaintainable code, etc.) - Q1, Q3
9. Latent S/W defects could cause the system to fail or not meet its requirements - T5, Q2, Q5
10. Late awareness (or lack of anticipation) of schedule, performance, cost and quality problems - T5, Q5, M2, M3
11. Software safety problem – A2+, A3
12. Executing faulty commands on a spacecraft – Q1, O2
13. Lack of robustness of functions supported by S/W – Q3, Q5, A4
14. S/W fails in a harmful manner – A1+, A2+
15. H/W and system failures compounded by inappropriate S/W responses – Q5, M4
16. Missing, wrong or extra software requirements - Q2, Q3, Q5, M2
17. Working with out of date requirements - Q2, Q3, Q5, M2
18. Failure to identify critical QA and V&V processes for S/W – T1+, T2+, A2+
19. Failure to identify critical contractor monitor points – M2, M3, O1
20. Can’t identify changes impacts (cost, schedule, functionality, etc.) – M2
21. Project progressing to the next phase of development before it is ready – M1, M3
22. Non-standard documentation and source code - M1, M2
23. Unable to effectively add personnel to an “in progress” project – T5, M1
24. Unable to make enhancements and changes to the S/W – Q1, Q2, M2
25. Un-reusable S/W products – Q2
26. Choosing the wrong/high risk contractor to develop software – M4, O1
27. Receiving wrong RFP responses with respect to S/W – M1, M2
28. Encountering a S/W error that wasn’t tested – M1, M4
29. Uploading faulty software to a spacecraft after launch – T5, M2, O2

**Other**

- O1-Support Contractor Mgt. (continuous assessment w/ RFP & SEB support from QA and V&V roles)
- O2-Mission Operations and Command Assurance (MOCA)

**Residual Risks (Missing Content)**

28. Encountering a S/W error that wasn’t tested (i.e., can’t test everything in a complex software product) – M1+, M4+
29. Uploading faulty software to a spacecraft after launch – T5+, M2+, O2+

**Appropriate Subset of Residual Risk Issue Relating to Selected Program Content**

10/21/98
<table>
<thead>
<tr>
<th>Mitigations (Risk Reduction)</th>
<th>Mitigations</th>
<th>Mitigations (Risk Reduction)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Use of an Automatic Code Generator</td>
<td>2 - Reusing high quality proven software products (req., design, code, and/or test cases) (*)</td>
<td></td>
</tr>
<tr>
<td>2 - Reusing high quality proven software products (req., design, code, and/or test cases)</td>
<td>6 - Lessons learned (*)</td>
<td></td>
</tr>
<tr>
<td>3 - Using Rapid Prototyping aspects of the software system</td>
<td>7 - Apply PACTS to critical functions (29)</td>
<td></td>
</tr>
<tr>
<td>4 - Simulation of software subsystem</td>
<td>12 - Cross training (29)</td>
<td></td>
</tr>
<tr>
<td>5 - Embedding Assertions in the code</td>
<td>14 - Incentivize contractor (*)</td>
<td></td>
</tr>
<tr>
<td>6 - Lessons learned</td>
<td>15 - Establish reuse requirements (28)</td>
<td></td>
</tr>
<tr>
<td>7 - Apply PACTS to critical functions</td>
<td>21 - Augmenting traditional V&amp;V with Formal Methods techniques (formal specification, model checking, animating specifications, and/or proofs) (28, *)</td>
<td></td>
</tr>
<tr>
<td>8 - Identify critical functions</td>
<td>10 - Use Complexity metrics (1, 4, 8, 20, 24, 28)</td>
<td></td>
</tr>
<tr>
<td>9 - Establish volatility metrics</td>
<td>18 - Use EVA metrics (1, 5, 10, 20, 26)</td>
<td></td>
</tr>
<tr>
<td>10 - Use Complexity metrics</td>
<td>19 - Standard documentation formats, reports (1, 4, 8, 20, 21, 22, 23, 24, 25)</td>
<td></td>
</tr>
<tr>
<td>11 - Early training</td>
<td>20 - Validation of auto code generator (1, 5, 8, 9)</td>
<td></td>
</tr>
<tr>
<td>12 - Cross training</td>
<td>21 - Augmenting V&amp;V with Formal Methods techniques (1, 3, 14, 16, 17, 28)</td>
<td></td>
</tr>
<tr>
<td>13 - Do regression testing</td>
<td>16 - Use TSRs (incl auto code gen) (1, 4, 8, 10, 16, 20, 21, 22, 24, 25, 28)</td>
<td></td>
</tr>
<tr>
<td>14 - Incentivize contractor</td>
<td>17 - Insight review of contractor SEI level (1, 5, 20, 21, 22, 26)</td>
<td></td>
</tr>
</tbody>
</table>

Note: * indicates a general risk reduction.