Design, Verification/Validation and Operations Principles for Flight Systems, Rev. 1

DMIE Document ID: DMIE-43913
Document Reference Number: D-17868

Section 0 - Introduction

Applicability

These principles apply to spacecraft, and to major payload/instruments. They apply to system contractor/partner as well as in-house/ sub-system project implementation modes.

Background

- JPL has a Quality Policy:
  - JPL will deliver products that meet or exceed customer expectations, while reducing cycle time and cost.
- JPL is committed to reduce the total cost of mission success (i.e., from design/development through flight operations) consistent with the faster, better, cheaper paradigm.
- The number of space missions at JPL is large and is expected to increase.
- Mission success must be achieved under conditions of tight budgets and short development schedules. These conditions create an intense demand for skilled, experienced personnel.

Activities To Achieve F-B-C Projects

There are several elements in the JPL plan to enable lower cost missions. JPL is aggressively pursuing several high-payoff actions.

- Collocating appropriate personnel to enhance work efficiency,
- Increasing interdependencies to reduce overall cost and promote teaming,
- Developing and maintaining model-driven design and test tools and facilities to reduce development time and cost,
- Improving access to key technical personnel to identify and resolve issues early,
- Increasing emphasis on mentoring to train personnel faster,
- Setting up an electronic parts replacement scheme to provide adequate stores,

- Documenting Develop New Products processes to standardize project development,
- *Documenting flight-proven system design, flight operations, and safety and mission assurance principles to guide project implementation.*

This last element is the subject of this document.

Scope

This document addresses the principles to be followed/ utilized in the formulation and implementation processes for JPL Flight Projects, including hardware and software design/development, margins, design verification, Safety and Mission Assurance and flight operations control and monitoring.

Additions/Updates

As technologies develop and design methods improve, and experience in the F-B-C environment increases, additions and revisions will be identified to these principles and passed through the same process as was taken to generate this initial set. The need for updating will be evaluated annually.

Categorization Of The Principles

These principles reflect JPL standards and the way of "doing business". The design principles are categorized into three sections:

Section 1 - General Principles

Section 2 - Detailed Principles

Section 3 - Flight Operations Principles

Adherence To Principles

These principles are **JPL Design/ Operations standards**, and each project is required to address them:

- Proposals responding to Announcement of Opportunities (AOs) will address exceptions (as known at that time) to the principles at the proposal Technical, Management and Cost Review.
- Project exceptions to the general principles (and any exceptions of the other principles known at the time of PIP preparation) will be documented, with rationale, in the Project Implementation Plan (PIP). This requirement is established in the JPL Project Planning Policy.
- Project exceptions to the detailed principles, and the flight operations principles will be addressed during the project lifecycle in the appropriate project reviews. The Project Review Plan will identify the appropriate reviews for each principle and how the principles will be addressed. These requirements are established in the JPL

Reviews Policy.

- Exceptions or changes to any of these principles identified subsequent to the PIP approval shall be documented by waiver. For tracking purposes a summary log identifying all waivers shall be included in an update to the PIP.
- All exceptions/ waivers to these principles shall be addressed at the Mission Readiness Review.
- For System Contract/ Partnering implementations, the project will obtain from the Contractor/Partner exceptions to the principles, in accomplishing the above.

Organization Of Principles

The principles within each section are listed more or less chronologically, as they would be considered and implemented during the course of a Flight Project implementation. Global principles are listed first, then design planning and requirements, design implementation, verification and validation, and finally flight operations.

Format Of Principles Descriptions

The principles are denoted as bold numbered expressions, and include a "**shall**" statement. The principles are covered by a numbered subject heading, such as "**1.6 Early Design Decisions**". Some principle descriptions are augmented by a supporting "***Rationale***" and/or an amplifying "***Note***" which are in italics. These statements generally apply to the principle described immediately above them, but in some cases may apply to several of the descriptions above them.

Glossary

AO	Announcement of Opportunity
ARR	ATLO Readiness Review
ASIC	Application-specific integrated circuit
ATLO	Assembly, Test, Launch and Operations
BER	Bit error rate
CBE	Current best estimate
CCSDS	Consultative Committee for Space Data Systems
CDR	Critical Design Review
COTS	Commercial Off-The-Shelf
CM	Configuration Management
DNP	Develop New Products (process domain)
EDAC	Error Detection and Correction
EDL	Entry, Descent, and Landing
EMI	Electromagnetic Interference
EPROM	Erasable Programmable Read-Only Memory
ESD	Electrostatic discharge
FMEA	Failure Mode Effects Analysis
FPGA	Field-programmable Gate Array
GPMC	Governing Program Management Council
HRCR	Hardware Requirements Certification Review
H/W	Hardware
IV&V	Independent Verification and Validation
PDR	Preliminary Design Review
PFR	Problem/Failure Report
PIP	Project Implementation Plan
PMSR	Preliminary Mission and System Review
POR	Power-on reset
PR	Procurement Requisition
PROM	Programmable Read-Only Memory
QA	Quality Assurance
RFP	Request for Proposal
RHU	Radioisotope heater unit
RTG	Radioisotope Thermoelectric Generator
SE	Support equipment
SRCR	Software Review (/) Certification Recird
S/W	Software
TMOD	Telecommunications and Mission Operations Directorate
TRL	Technology Readiness Level

Section 1 - General Principles

1.1 Priorities

1. Safety of people (project personnel, flight crew, public) shall be the paramount requirement. Hence, safety requirements and compliance thereto shall be shown to be not compromised in trade-offs against the other project parameters.
   Rationale: *Safety of personnel must not be compromised while achieving mission objectives.*
2. The project shall focus on applying prudent engineering and risk management decision-making to achieve mission success within the cost and schedule constraints, sacrificing performance if necessary. Consistent with this, the project ordered priorities shall be safety, reliability, cost, schedule, and performance.
3. Project shall also prioritize/ weight mission/system competing requirements, with rationale, to guide design and implementation trade-offs.
4. The Project shall provide in the appropriate planning documents a description regarding how priorities will be used to guide the resolution of technical/programmatic issues as well as guide the design implementation.

1.2 Develop New Products (DNP)

1. The design shall use demonstrated and validated Develop New Products (DNP) design modeling/simulation methodologies and processes.
   Rationale: *To reduce cost and speed development.*

1.3 Flight Hardware Logistics Program (FHLP)

1. The Flight Hardware Logistics Program shall be used whenever it can provide schedule or financial benefits for a project. The planned usage of equipment from this program shall be included in the project proposal and planning documents.
   Rationale: *Provides projects and programs with opportunities to save time and/or money by providing an inventory of materiel, and brokering and managing consolidated procurements for materiel among projects*

1.4 Mission Data System (MDS)

1. Compatibility with the Mission Data System (MDS) approach for design/development of flight and ground system software, test software, and flight scenario (sequence design and verification) software shall be a major consideration.
   Rationale: *Use of the MDS complies with the long-term JPL objectives and strategy for concurrent operations of many small missions.*

1.5 Modeling/ Simulation

1. Modeling and simulations shall be used early and often to develop and evaluate designs. The models shall be realistic taking into account real mechanical/electrical configurations, intended operational conditions, material properties, etc.
2. Expectations/ predictions for the models/ simulations used, and their fidelity limits, shall be documented.
3. Models/simulations shall be test-validated.
4. Modeling/ simulation outputs shall be assessed and compared with predictions and system requirements.

1.6 Make Early Design Decisions

1. Projects shall identify and maintain a "top ten" list of required design decisions and milestones.
   Rationale: *It is better to make a few "questionable" decisions that keep the design process moving forward rather than delay decisions to make the design "perfect". Remember, "better is the enemy of good;" hence, avoid the temptation to make the design better if it is already good enough.*
   Note: *This principle is enabled by early definition and approval of requirements and capabilities*

1.7 Design to Requirements/Capability

1. Mission and system requirements, and intersystem capabilities and interfaces, shall be baselined (approved by the project), and a functional design identified which can satisfy the requirements, as early as feasible, but by Project PDR at the latest.
   Note: *Desires/goals may be considered if acceptance provides high payoff at low cost/risk or significantly reduces/avoids future cost/risk.*
2. The design process shall consider existing capability and the cost effective use of inherited (flight and ground) designs, H/W, S/W, SE, etc., as a major cost/risk reduction.
3. Designs shall use commercial off the shelf (COTS) functionality where it is feasible and reduces cost/risk; especially consider COTS for Prototype, EM and Support Equipment (SE).
4. Particular attention shall be given to Red Flag PFRs/significant PFRs and ISAs relating to the existing capability.
   Rationale: *Identifies incompatibilities between previous usage and current mission requirements.*
5. The design process shall consider the use of new concepts/advanced technology when it is needed to meet the priorities, provides or preserves prudent margins, or is identified as a requirement to enable future missions.
6. New technologies/concepts shall be identified in the project implementation plan with the associated risks, including actions being taken to maximize prospects for success, addressed. Use of technologies whose assessed maturity are less than TRL level 6 shall be justified in the PIP.
7. Project shall conduct as a part of the transition to phase B a capabilities vs. requirements review. Compatibility of inherited capabilities, and adequate qualification of new technology shall be demonstrated with respect to the system functional, performance, and environmental requirements at the review. This review can stand alone or augment/support the PMSR.
8. Design inheritance reviews shall be held as early as practical in phase B to assess the system compatibility of inherited functionality with the defined level 3 and 4 requirements. At a minimum address:
   mass, power, performance, interfaces, requirements changes, design analyses, environment qualification (hardware and software), test history, support equipment compatibility, problem logs/PFRs and waivers from previous usage, parts reliability, material/process changes, parts/component availability and margins.
9. Operating characteristics and requirements drivers from inherited/ existing flight systems elements on the flight operations system design and vice-versa shall be identified in the preliminary design activity and reviewed at the flight system PDR.
   Rationale: *To identify requirements feasibility and drivers, and to assess intersystem, and flight system/ subsystem design compatibility with them, such that existing capability costing assumptions can be validated, and costly later changes will be avoided.*

1.8 Standards

1. Industry and JPL Standards (H/W and S/W) shall be considered in all areas of the design to reduce cost/risk, e.g., Consultative Committee on Space Data Standards (CCSDS). Where deviation from standards is necessary, and risk is consciously accepted, these risks shall be included in the Significant Risk List.

1.9 Risk -Based Design Trade-Offs and Margin

1. Trade-offs based on balancing risk shall be used in design/ development/ mission operations decision making and be consistent with the project's principle for priorities, particularly safety.
   Rationale: *A balanced risk approach improves the prospects for success within technical/programmatic resources. Trade-off studies can be used to identify effective use of programmatic and technical resources in varying combinations, enabling proactive risk management to balance/reduce overall project risk.*
2. Hardware/ Software trades shall be performed early in the project using risk as a metric.
3. Design and programmatic resource margin requirements (e.g. mass, power, budget, schedule) shall be established early in the project. The usage of these margins to effectively solve problems and mitigate risk shall be a part of the margin management planning.
4. The actual usage of the margins shall be assessed and reported against the plan regularly throughout the life-cycle. Corrective action shall always be considered when actual usage deviates significantly from plan.
5. To maintain this balance, Project shall consider accepting, when prudent, the least unsatisfactory trade-off solution for problem resolution.

1.10 Single Failure Tolerance/ Redundancy

1. No credible single failure of any electrical, mechanical or electromechanical element shall result in loss of the entire mission.
   Note: *Redundancy may be used to provide protection against potential single point failures. Redundancy may be implemented as block or functional redundancy.*
2. Where block redundancy is used, cross-strapping circuitry shall be subjected to Failure Mode Effects Analysis to demonstrate intended reliability improvements.
   Note: *Cross strapping adds significant cost, possible "sneak path" failure modes, and increases system complexity, requiring more extensive fault analysis, system and subsystem testing, and more test time to acquire operating hours and characterize the cross-strapped configurations.*
3. A potential single point failure **exemption** list shall be developed, (e.g., primary structure).
4. During development a list of potential credible single point failures shall be developed, maintained and reported at PMSR, PDR, CDR, ATLO START and Launch.
5. The list of accepted potential single point failures shall be communicated to the flight operations team. Particular attention shall be given to those items where the risk mitigation plan requires flight operational actions.
6. All identified potential single point failures shall be addressed at the Pre-Ship Review and the Mission Readiness Review.
7. Use of single-string design may be considered if risk can be demonstrated to be acceptable.
   Note: *Engineering subsystems are generally made redundant for long missions. Missions may consider single string designs based on historical failure data, statistical trade studies, design robustness and consequences of mission failure. Science Instruments can generally be single string. Projects that adopt a single-string operational approach for critical events should do so with special attention to the use of functional redundancy and control algorithm robustness (e.g. EDL.)*

1.11 Nuclear Materials

1. The design shall avoid the use of nuclear materials, (e.g., RHUs, RTGs) unless they are essential to mission viability or overwhelmingly cost-effective.

1.12 Design Fallback Options

1. Design descope or fallback options shall be identified early in design conceptualization.
   Rationale: *High risks (e.g. using new technology) may be carried longer in the implementation if descope options are available.*

   Note: *The impacts on the design performance and effort resulting from exercising these options must be understood and acknowledged.*
2. Trigger-events/dates shall be identified in advance and adhered to.
   Rationale: *This facilitates decisions as late as possible while still retaining the benefit of descoping.*

1.13 Safety and Mission Assurance

1. The project shall plan early in the formulation phase for adequate safety and mission assurance activity, and shall identify the responsibilities of the participating organizations in tailored Safety and Mission Assurance plans. These plans shall define the project's implementation of the following JPL processes:
   - Mission Assurance and Independent Assessment
   - System Safety
   - Reviews
   - Risk Management
   - Reliability Engineering
   - Quality Assurance
   - Electronic Parts Engineering
2. Assurance engineering shall be integrated and concurrent with the design activity throughout the project life cycle.
3. Project quality assurance provisions shall be flowed down to all project acquisitions.
   Rationale:
   - *Proposals should reflect S&MA approach to customer, and assurance engineering can be involved in the earliest design decisions.*
   - *Avoids redesign resulting from after-the-fact MA review, and resolves product quality issues as they arise.*
   - *Communicates the mission assurance program with the project, and provides acknowledged infusion of S&MA into development processes.*

1.14 Design Margins

1. Design margin requirements shall be established consistent with design maturity, mission environments, and consider potential changes due to environment and mission/system design uncertainties, and "don't know-don't knows".
2. Design margins shall be robust enough to accommodate design uncertainties and enable design changes with minimal system-wide "ripple effects".
   Rationale: *Robust margins enable design and programmatic trades to be made effectively and rapidly without lengthy studies, thereby preserving programmatic resources (budget and schedule).*
3. Design margins shall be managed and traded at the highest possible level in the mission or system (e.g. performance vs. available power, allocations of timing uncertainties.)
   Rationale: *If margins are locally traded, artificial constraints can be created which unnecessarily reduce the system capability to achieve the prime mission/science objectives* ***OR*** *cause non-productive work.*
4. Robust margins in system resources shall be available for flight operations.
   Rationale: *Sufficient margins in system resources such as power, thermal range, telecom, memory, timing, bandwidth, pointing, and delta-V improves operability by enabling operators to effectively accommodate differences in flight conditions from predicted, and by maximizing response capability to anomalies while preserving mission return.*

1.15 System Performance Allocations

1. The project shall identify, and allocate nominal values and uncertainties to the distributed contributors to system performance (e.g. pointing error contributors) as early as possible. Driving system performance contributors shall be identified and included in the project risk assessment.
2. Estimates of nominal and uncertainty values shall be updated as often as needed, and specifically reviewed at major design reviews (e.g. PDR, CDR).

1.16 Combining System Performance Contributors

1. The approach (e.g. linear sum, RSS, confidence level, etc.) for combining system performance contributors' nominal values plus uncertainties shall be defined for each performance measure.
2. The combining approach shall take into account the dependence/ coupling of the contributors, the nature of the uncertainties (systematic or random) and the uncertainty distributions (Gaussian, uniform, etc.).
3. Specific unallocated margin, relative to the performance requirement, shall be identified and maintained in this combining process.

1.17 JPL Lessons Learned/NASA Alerts

1. The design shall be reviewed early in the formulation process, and at appropriate points in the life-cycle, by the engineering team against the JPL/ NASA Lessons Learned data base, NASA/JPL Alerts, etc. Items of potential applicability to the project shall be identified and dispositioned.
   Rationale: *Important "lessons" can be drawn from past events, which have applicability beyond the original event, which can preclude recurrence of faults/failures, and enable early and cost-effective changes. Some examples of past troublesome areas are:*
   - propulsion system contamination
   - cabling (e.g. wire treatment, open pins, insulation
   - power converter design
   - micro-meteoroid modeling and protection
   - deployments (e.g. booms, covers)

1.18. Project Risk Assessments

1. The project shall perform, with appropriate independent assessment support, a total mission risk assessment, at inception of project, and in reviews as defined. These assessments shall be documented in the appropriate project planning documents.
   Rationale: *To ensure JPL and customers are informed of risk to program/project success, and to provide independent assessment back to project to enable possible mitigation approaches outside the project's sphere of influence.*
2. These assessments shall specifically identify and address risks to project and program objectives.
3. Risk assessments shall specifically include margin assessment as one of the risk metrics.

1.19 Closed-Loop Failure Reporting and Flight Team Awareness

1. The JPL electronic problem log/PFR System shall be used. If an appropriate contractor's system is used, significant and red-flag PFRs (at a minimum) shall be recorded in the JPL PFR system as well.
   Rationale: *Uniformity of describing and reporting problems, and consistent reference capability enables cross-project understanding of risks and implications of the issues.*
2. The project shall establish and use a concurrent engineering process involving the appropriate project team members (e.g. designers, systems engineers, assurance, test and operations engineers) to close problems in a timely and confident manner. The Project Manager shall disposition the acceptance of Red Flag PFRs and address them at the Pre-Ship and Mission Readiness Reviews.
3. Red Flag PFRs and significant PFRs shall be compiled and forwarded routinely to the flight operations team, preferably at or before the beginning of Flight-Ground System end-to-end testing.
   Rationale: *To make the flight team aware of those pre-launch problems that may pose a significant threat to flight operations activities.*

1.20 Peer Reviews

1. Projects shall use independent peer oversight/ review prior to design reviews (PDR, CDR, etc.). Peer reviews shall include intra-system reviews (e.g. Entry, Descent and Landing, Fault Protection.) These reviews shall be implemented using the procedure "Planning and Implementing Peer Reviews", in DMIE.
   Rationale: *Reviews provide early detection and correction of deficiencies, and provide periodic assessments of progress against plan. Use of experts from outside the project team improves the activity. Peer reviews allow discipline-specific penetration into the details of design and implementation issues. Reporting the findings to a subsequent project review provides the review board with essential detailed insight otherwise not available in the formal review presentation format.*
2. PDR or CDR shall address the findings and actions from the peer reviews.
   Rationale: *Peer reviews properly focus on the adequacy and characteristics of the detailed design of elements of the system. In order to maintain compliance with mission and operations customer expectations, timely independent review of results of the peer reviews against systems requirements, concept of operations and mission design is necessary and effective.*
3. Peer Reviews shall be recommended/ considered by line management or project as needed to deal with special topics or issues as they arise.

1.21 Testability

1. The design shall be implemented so that hardware testing, including in-situ troubleshooting activities, can be effectively performed at the subassembly (board), assembly, subsystem and system level.
2. The design shall enable software testing at unit, module, subsystem test bed and system test bed levels to incrementally verify functionality/operability.
3. The software design shall include self-test and built- in test routines to test operation and permit timely fault diagnostics.
4. The software self-test and built-in test routines shall be removable for flight. If not removable, the test routines shall not cause flight hardware damage or interfere with proper operation of the flight software if inadvertently executed in flight.
5. The design shall enable "early and often" testing throughout development.

1.22 Accessibility

1. The design shall provide sufficient accessibility to permit hardware rework/ verification in a reliable, efficient manner without adding unwarranted hardware risk.
2. The design shall avoid the use of "blind" mating of electrical connectors.
3. Sharp corners or edges shall be avoided in the flight system design.
   Rationale: *Precludes injury to personnel, or damage to hardware caused by snagging of garments.*

1.23 Test Beds

1. The number and type of test beds, including S/W-only test beds, and support equipment (hardware and software) shall be identified and provided for early in the development plan.
   Rationale: *Multiple test beds enable concurrent testing to be done at various stages during the development cycle. Multiple test beds enable a "build a little, test a little" design approach for early software or hardware/ software problem identification and resolution. S/W only test beds enable early software and/or flight sequence problem identification and resolution prior to committing to the more expensive and often time-critical use of the hardware system test beds.*
2. Test bed fidelity shall be maintained. Differences (H/W and S/W) from flight shall be documented and maintained. Simulation models shall be validated by test, using sufficient parametric variation in the simulations to ensure the existence of adequate margins, when system-level flight system verification is performed on a test bed.
   Note: *During component selection, system engineers should consider the modeling feasibility and effort as an extremely important selection criteria.*

1.24 Test and In-flight Protection of Flight Hardware

1. Flight hardware interfaces with ground handling and test equipment shall be designed with protective overvoltage/ overcurrent or overpressure, etc. devices.
   Rationale: *Precludes test operator/test equipment or environmentally-induced (e.g. lightning) damage or degradation to flight hardware.*
2. The system/hardware and software developers shall provide to the flight operations team a set of operating flight rules, including health maintenance rules and rules for life -limiting elements that ensure the health/safety of the flight system.
3. System developers shall be involved in conducting flight operations, particularly in the early operations activities following launch.

1.25. Systems Validation

1. The Project shall establish a plan for providing demonstration that the integrated project systems will accomplish the intended mission and effectively satisfy the customer's goals and objectives.
2. The Project shall perform a Mission Design Verification Test (MDVT) to validate the end-to-end Spacecraft - Deep Space Network - Ground Data System capability of the mission systems to accomplish mission objectives.

   Note: *System Validation includes:*

   - establishing that the requirements identified through mission synthesis to the implementing systems, traced down to the lowest levels of implementation, meet customer needs.
   - demonstrating, through the Systems Design Verification activity, the confidence that the requirements are met, and the inter-operability of the hardware and the software.
   - demonstrating through end-to-end integrated systems analysis, demonstration, acceptable inter-operability and robustness of the systems,
   - demonstrating through operational readiness testing, that the people and procedures function effectively in flight-like operations environments, including all voice, command, telemetry and decision paths, and in a realistic mission timeline.

1.26 Design Verification

1. The Project shall establish a systematic, comprehensive system design verification plan showing how all system requirements compliance will be demonstrated.
2. "Test as you fly and fly as you test" (e.g. using flight sequences, flight-like operating conditions, and the same software functionality) shall be the system verification philosophy. Where testing is not possible, verification shall be demonstrated by independent analyses.
3. The design verification plan shall at least provide for nominal and off-nominal end-to-end system verifications, environmental verifications, fault protection, flight sequence and cross-system verifications.
4. Appropriate system-level stress testing (beyond normal design verification level) shall be performed to determine capability boundaries and demonstrate robustness of the end-to-end systems design, in order to assure health/ safety and provide confidence in successful completion of mission critical activities. Stress testing shall consider testing, for example, single faults that cause multiple-fault symptoms, occurrence of subsequent faults in an already faulted state, etc.
5. The design verification plan shall also provide for early system functional and performance verifications. In particular, system level verifications shall include testing of appropriate flight sequences under both nominal and simulated faulted conditions, verifications of interfaces with the Launch Vehicle, the Deep Space Network, the Ground Data System, and other project-unique interfaces. The plan shall require a system level electrical "plugs-out" test using the minimum number of test equipment connections.
6. In addition to usual real-time data analysis, comprehensive non real-time analysis of test data shall be planned to identify problems and enable early resolution with minimal cost/schedule impact.
7. Hardware and software verification shall be planned during the formulation phase.
8. Testing shall be the primary method for design verification. If test verification is not practical or

appropriate, other methods such as modeling/simulation using test-verified models/simulations, analysis and inspection methods shall be specified and used. Results of verification by simulations and/or analyses shall be independently reviewed.

9. Verification by visual inspection of mechanical clearances and margins (e.g. potential reduced clearances after blanket expansion in vacuum) shall be performed on the final as-built hardware.
Rationale: *To verify the adequacy of thermal blanket clearances, etc. before and after environmental test, handling, etc.*

10. Verification of all deployable or movable appendages and mechanisms shall include full-range articulation.

1.27 Use of Engineering or Prototype Hardware

1. Engineering or prototype models shall be as identical as practical to the flight units in the functionality being tested. When used to validate/qualify a design, the model shall be tested to at least the same levels and in the same manner as the flight unit.

2. If engineering or prototype models are intended to be possible future flight spares, the plan for this usage shall be established early in the design concept development.

3. To enable use of engineering or prototype models as flight spares, appropriate actions shall be taken to ensure hardware safety, reliability, and functionality.

1.28 Use of Protoflight Hardware

1. Protoflight hardware (hardware intended to be flown for which there is no direct qualification heritage) shall be validated/ qualified to the following conditions:

- Themal - Qualification levels and durations
- Dynamic - Qualification levels, but Flight Acceptance durations
- EMC/ Magnetics - Qualification levels

1.29 Critical Hardware Power On/Off Cycling

1. In-flight routine power cycling of critical hardware for power margin management purposes shall be avoided, *unless cycling is essential to mission viability and the risk is demonstrated to be acceptable.*

1.30 Critical Sequence Telemetry/Monitoring

1. The design shall provide the capability for simultaneous real-time transmission and on-board storage of mission critical sequence (e.g., launch, fly-by science, orbit insertion, entry/descent and landing, etc.) data. Stored critical data shall be protected from loss in the event of selected anomalies, (e.g., transient power outage) and shall be transmitted to Earth as soon as practical.
2. Mission critical event (e.g., Launch Vehicle separation, deployments, etc.) and deployables verification shall be available via real-time telemetry.

1.31 Earth Orbital Debris

1. Orbital debris safety considerations shall be addressed during the project formulation phase and during the implementation phase.
2. Orbital debris from launch vehicles, spacecraft, instruments or components thereof (e.g., launch vehicle 2nd or 3rd stage, instrument covers) shall be limited, as much as practical, by employing prudent design and flight operations techniques, as appropriate.
   - The design and flight operations shall employ debris-limiting options (e.g., propellant depletion burns, cover release inhibits) considering normal and off-normal operations, and certain anomalous events (e.g., explosions, breakups, or collision with other debris).
   - Identification of orbital debris sources, potential hazards and a debris-limiting assessment shall be presented at the PMSR. Functional design implementation shall be reviewed at the Project PDR and finalized at the CDR.

   Rationale: *Limit the proliferation of debris that may be a safety threat to personnel or space vehicles (current and future) generated by orbital debris.*
3. Earth orbiting spacecraft shall be designed (wherever possible) with capability to be de-orbited reliably at the end-of-mission.

1.32 Telecommunication Telemetry/Command Capability

1. Telemetry and command capability shall be available throughout the mission in normal cruise pointing attitude, and during special cruise phase mission/system activities (e.g., long duration Deep Space Trajectory Correction Maneuvers, propulsion mission-critical pyro device actuations).

   Rationale: *To provide "real-time" monitoring of activities and enable ground contingency commanding, if necessary.*

1.33 "Keep-it-simple" Design Philosophy

1. Designs shall employ a "keep-it-simple" philosophy (i.e., straight-forward designs) to reduce risk/cost, to enable easy implementation, design verification and flight operational usage (e.g., where appropriate, passive antenna coupling vs active switching).
2. Use of "complex" design implementations shall be avoided. Added complexity shall be justified to be essential to meet mission requirements/constraints.
   Rationale: *To maximize the prospects for safe, reliable operation.*

1.34 Hardware/ Software System Design and Verification

1. Standards shall be utilized in defining HW/SW and SW/SW interfaces between the flight systems and the ground, between flight systems and within a flight system; e.g., CCSDS for telemetry and command.
2. System/mission requirements shall be traceable to the project-level requirements, and detailed requirements on hardware and software elements, and interfaces between them, shall be traceable to the system/mission requirements.
   Rationale: *To ensure completeness & correctness of critical requirements, in order to use requirements to accomplish software and system validation.*
3. Mission scenarios shall be generated early and used to enable effective hardware/ software functionality allocation. They shall be maintained current and used to guide the design, integration and test activities at all levels.
4. The number and type of interfaces employed in the design of the flight software shall be minimized.
5. Test/diagnostic code shall be designed and incorporated into the software early so that problem resolution can be done rapidly and easily at element and flight system level, and adapted by the flight operations team.
6. Fault case issues shall be addressed and solutions incorporated into the design as early as practical during the design cycle. Fault protection software shall be specified in the systems engineering process to handle all credible flight system single -fault scenarios.
7. Prior to computer design/procurement, analysis shall be employed to estimate the amount of throughput and memory required to meet the project needs. Procured or designed processing components shall exceed estimated requirements by at least a factor of 4.

1.35 Mandatories List

1. No later than 9 months before launch, the Project shall develop and maintain a prioritized list specifying the mandatory ATLO tests and mission operations tests/ products that must be completed to commit to the launch. Changes to the list, or test/ product shortfalls, shall be reviewed/ approved by the Project Manager.
   Rationale: *Permits Project to focus on the essential work and make the most effective use of personnel, schedule, and budget resources.*

1.36 Projects Budget and Schedule Reserve

(A) Budget Reserve

1. Budget reserves shall be planned and managed.
2. Budget reserves shall be assessed and reported periodically and at major Project milestones.
3. When assessment of reserves results in less than the specified level, an action plan shall be developed for approval by the Director For.

Definitions:

Budget Reserve = {Unencumbered Budget Reserve/Estimated Cost-to-Go} x 100%
Total Budget = Estimated Cost-to-Go + Unencumbered Budget Reserve

Flight Missions Flight Experiment Projects Budget Reserves (%)	Step-1 Proposal &/or A-to-B Transition Proposal Response to AO	Step-2 Proposal &/or A-to-B Transition Proposal Response to AO	Project PDR &/or B-to-C/D Transition Instrument/Payload PDR	Project CDR Instrument/Payload CDR	Start of ATLO Start of Instrument/Payload I&T	Ship to Launch Site Deliver to ATLO
	30%	30%	25%	20%	20%	10%

Example:

Unencumbered development (phase C/D) budget reserve = 25% of the estimated development (phase C/D) Cost-to-go at the Project PDR, or equivalently, the phase B-to-C/D transition.

Notes:

- *Mission operations (phase E) budget reserve = 15% at the start of phase E.*
- *Budget reserve for operations during extended missions = 5% (assumes carry forward of any unused budget reserve is allowed).*
- *Cost-to-Go includes the funded schedule margin, but excludes the launch vehicle costs.*
- *Developments, if any, deferred to phase E, require appropriate budget reserve separate from that identified herein for mission operations.*
- *Budget reserves for phase E in early project life cycle cost estimates may be more than those specified herein, in order to account for the uncertainty in operations complexity.*
- *Budget reserves less than those specified may be appropriate in certain cases. For example, developments with a high degree of directly relevant inheritance, or where there has been a large investment during formulation to retire implementation risk.*
- *Budget reserves more than those specified may be appropriate in certain cases. For example, where development of low TRL enabling technology is necessary, or where de-scope options represent significant mission degradation, or where the other margins (schedule, technical) used to manage risk are at the lower limits of their acceptable ranges.*
- *The distribution of the budget reserve is important in the management of risk. It is essential that adequate budget reserve be available in all fiscal years to allow a management response to the threats to a successful outcome.*
- *The result of Project-specific tailoring, if any, of the budget reserve principle is documented in the approved Project Plan.*

(B) Schedule Margin

1. Schedule margins shall be planned and managed.
2. Schedule margin shall be assessed and reported periodically and at major Project milestones.
3. When assessment of margin results in less than the specified level an action plan shall be developed for approval by the Director For.
4. Schedule margin shall be identified in the Project Plan.

Definitions:

Schedule Margin =	No planned activities, but funded schedule
Total Schedule =	Planned activities + Schedule Margin
Schedule Margin Rate =	Schedule Margin/(Planned Activity +Schedule Margin)

Flight Missions	From start of implementation to delivery to ATLO	From start of ATLO to ship to launch site	From delivery to the launch site to launch
Flight Experiment Projects	From start of implementation to delivery to instrument/payload I&T	From start of instrument/payload I&T to delivery to ATLO	N/A
Schedule Margin Rate	1 month/year	2 months/year	1 week/month (2.8 months/ year)

Notes:

- Schedule margins less than those specified may be appropriate in certain cases. For example, developments

with a high degree of directly relevant inheritance, or where the impact of missing the delivery milestone is tolerable. The specified schedule margins assume impacts resulting from flight mission launch delay and flight experiment Project late instrument/payload delivery are significant.

- Schedule margins greater than those specified may be appropriate in certain cases. For example, where the development of low TRL enabling technology is in series with delivery, or where cryogenic system testing is required before delivery.
- Management of schedule use becomes increasingly difficult in progressing through the development lifecycle as degrees of freedom decrease, constraints increase, and time to solve problems becomes short. The specified schedule margins recognize this experience by requiring ample margins for later activities where the schedule for resolution of problems is under less direct control of the Project management.
- The result of Project-specific tailoring, if any, of the schedule margin principle is documented in the approved Project Plan.

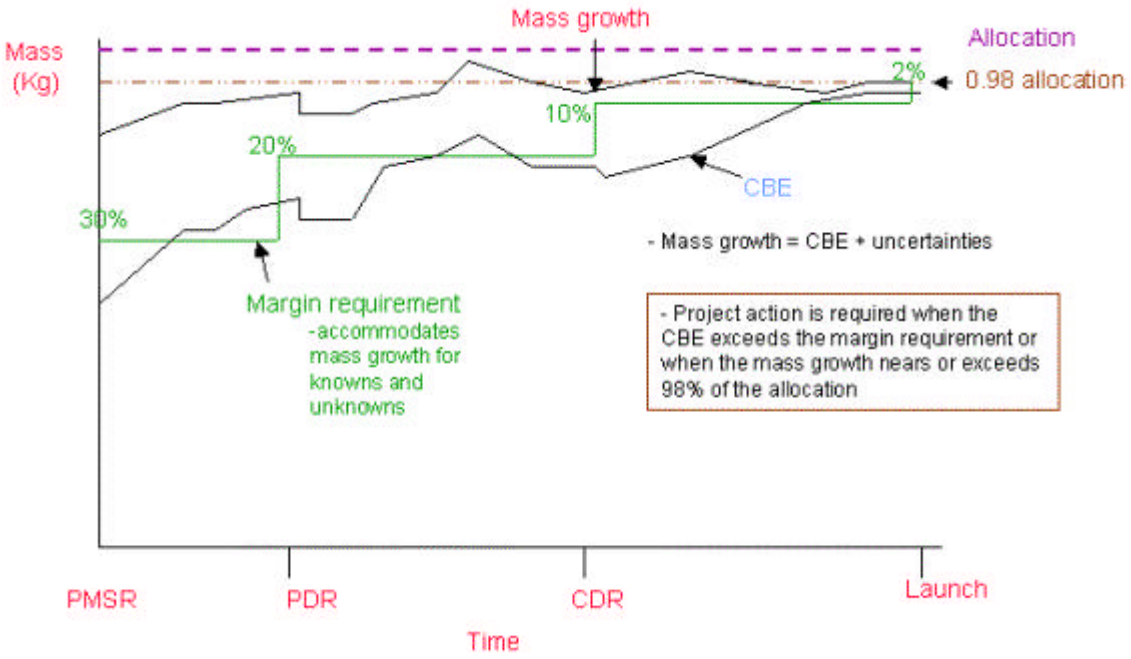
Section 2 - Detailed Principles

2.1 System Mass and Power Margins

Because of the ambitious nature (technical and programmatic) of JPL missions, aggressive, balanced risk management is necessary to enable success. Therefore, it is prudent to have ample mass and power resources to account for and accommodate uncertainties and expected growth. Furthermore, ample mass and power resources in conjunction with ample funding resources provide flexibility to resolve developmental and operational issues, and enable timely, balanced risk management decisions without having to perform time -consuming trade studies to micro-manage every kg. and watt. For example, projects can use funding to mitigate mass and/or power growth, or use both funding and mass to mitigate power growth, or use mass and power to preserve budget and schedule.

The detailed design principles for mass (2.1A) and power (2.1B) margins are based on a review of actual growth histories (from about phase B start to ATLO start) for several flight projects. These data suggest that total mass and power growth, from knowns and unknowns (items that became known only as the design was being implemented) ranged from 20% to 48%, with most in the range of 25% to 40%. Factors affecting growth included mission/system design changes, design complexity, amount of inheritance, amount of new technology/ concepts, quality/fidelity of early estimates, and funding available.

A mass metrics versus design life-cycle maturity chart (Figure 1) is provided to graphically illustrate margin, current best estimate (CBE), allocations and growth. A similar chart can be generated for power, etc.



(A) Mass Margins

Definitions:

Margin = Allocation - Current Best Estimate (CBE);

% Margin = $\frac{\text{margin}}{\text{allocation}}$ X 100

- The above % margin definition shall be used by all projects in both the formulation and implementation phase.
- Allocation is defined as the capability from the launch vehicle.
- CBE is defined as: best estimate taking into account everything known.
- To improve the prospects for meeting the actual allocation, a reduced allocation may be used in the above management algorithm, which subtracts a reserve of a few percent (e.g., <5%) of the allocation as a margin management reserve.

1. Positive Margins shall be maintained throughout the development cycle.

2. The following algorithm, based on mass growth history, shall be used to estimate mass growth:

- **New designs** shall use **30%** *or mo*re growth from the PMSR depending on the nature, maturity, amount of new technology/ concepts, and complexity of the design.
- **Inherited designs** shall use **15%** *or more* growth from the PMSR depending on the outcome of inheritance reviews.
- **Inherited hardware** shall use **10%** *or more* growth from the PMSR depending on the outcome of inheritance reviews.
- **Inherited hardware** shall use **2%** growth from the PMSR if hardware is totally known to be without change, and is "build-to-print". Any change to these conditions should be evaluated and a larger growth percentage applied.

3. Experience indicates there will likely be significant growth to deal with knowns and unknowns. Adequate margin shall be provided to accommodate growth. Hence, Spacecraft system level mass margin shall be at least 30% at the PMSR, 20% at Project PDR, 10% at CDR, 5% at ATLO Readiness (ARR) and 2% at launch or as set by the project manager.
4. Required margin curves shall be generated for mass to assess status throughout the design life cycle, e.g., PMSR, Project PDR, CDR, ATLO start, Launch. Margins can decrease as design maturity increases (less uncertainty).
5. The entire spacecraft mass shall account for on-board propellants. The propellant load shall be sized to provide the required delta velocity for the total mass allocation.
6. Mass CBEs and mass growth shall be reported and compared with the required margin curves to assess margin status periodically (at least quarterly) and at implementation design reviews. Mass growth estimates shall be combined in a linear, RSS'd or combination thereof, depending on dependence/ coupling. Monthly mass reporting shall be considered, where appropriate.
7. Significant deviations from the mass margin requirements shall be accompanied with rationale and recovery options/impacts.

(B) Power Margins

Definitions:

Margin = Allocation - Current Best Estimate (CBE);

$$\% \text{ Margin} = \frac{\text{margin}}{\text{allocation}} \times 100$$

- The above % margin definition shall be used by all projects in both the formulation and implementation phase.
- Allocation is defined as the capability from the power source. Where the capability degrades with mission duration, the allocations should be based on specified (end-of-mission) performance
- CBE is defined as: *best* estimate taking into account everything known.
- To improve the prospects for meeting the actual allocation, a *reduced* allocation may be used in the above management algorithm, which subtracts a reserve of a few percent (e.g., <5%) of the allocation as a margin management reserve.

1. Positive Margins shall be maintained throughout the development cycle.
2. The following algorithm (based on history) shall be used to estimate design growth:
   - **New designs** shall use **30%** *or more* growth from the PMSR depending on the nature, amount of new technology/ concepts, maturity and complexity of the design.
   - **Inherited designs** shall use **15%** *or more* growth from the PMSR depending on the outcome of inheritance reviews.
   - **Inherited hardware** shall use **10%** *or more* growth from the PMSR depending on the outcome of inheritance reviews.
   - **Inherited "use-as-is" hardware** shall use **2%** growth from the PMSR. Any change to these conditions should be evaluated and a larger growth percentage applied
3. Experience indicates there will likely be significant growth to deal with knowns and unknowns. Adequate margin shall be provided to accommodate growth. Hence, Spacecraft system level power

margin for cruise, mission critical, and safing modes shall be *at least* 30% at PMSR, 20% at Project PDR, 15% at CDR, and 10% at ATLO Start (ASR).

4. Required margin curves shall be generated for power to assess status throughout the design life cycle, e.g., PMSR, Project PDR, CDR. Margins can decrease as design maturity increases (less uncertainty).
5. Power CBEs and power growth shall be reported and compared with the required margin curves to assess margin status periodically (at least quarterly) and at implementation design reviews. Power growth estimates shall be combined in a linear or RSS'd or combination thereof, depending on dependence/ coupling. Monthly power reporting shall be considered, where appropriate.
6. Significant deviations from the power margin requirements shall be accompanied with rationale and recovery options/impacts.
7. At launch, there shall be at least 10% predicted power margin for mission-critical, cruise, and safing operating modes, to accommodate in-flight operational uncertainties (e.g., unexpected increases in electrical load consumption or different than pre-launch planned usage profile and/or less than pre-launch expected power source output), improve the prospects for successful completion of mission critical activities, and reduce the likelihood of an under-voltage fault condition.

2.2 Flight Software Margins

1. Prior to computer design/procurement, analysis shall be employed to establish margins for critical performance resource parameters such as CPU speed, control cycle rates, interrupt rates and durations, communications bandwidth, random access memory (RAM) and erasable programmable read-only memory (PROM and EPROM). Analysis results are documented as the Current Best Estimate (CBE). A development shall observe the following experience-based guidelines for margin at critical development milestones:

   - At computer selection, total capability to be: 400% of CBE
   - At implementation start (start of Phase C/D): 60% margin
   - At launch: 20 % margin*

   where

   Margin = Total Capability – CBE
   %Margin = 100x(Total Capability-CBE)/(Total Capability)

   * To accommodate post-launch fixes, new capabilities, and to maintain adequate in-flight operating margins
2. The flight software shall be designed to support measurement of computing resources, such as throughput and memory.

   All margin and performance estimates are considered speculative until measured. External instrumentation is recommended.
3. CBE's for identified margins shall be tracked continuously and reviewed at least quarterly as well as at PDR, CDR, ATLO start, and Launch. Margins shall be re-examined in conjunction with proposed significant design changes.
4. Significant deviations from the margin requirements shall be accompanied with rationale and recovery/options impacts.
5. Flight software shall accommodate both nominal hardware inputs (within specifications) and transient off-nominal inputs from which recovery may be required.

2.3 Power-On Reset (POR) State/Toggle Commands

1. At prime power turn-on or recovery from a power under-voltage condition, each subsystem shall autonomously configure to a unique, unambiguous, safe, system compatible state.
2. A POR occurrence shall be unambiguously identifiable via telemetry.
3. To reduce uncertainty of knowledge of spacecraft state, toggle commands shall be avoided.

2.4 Fault Protection/ Flight Team Commandability

1. The fault protection system design shall be in-flight-commandable to permit changing the state of enable/disable parameter and other pertinent parameters, e.g., threshold and persistence values. The status of these parameters shall be telemetered and made available for timely flight team use.

2.5 System Fault Recovery State Response

1. During non mission-critical cruise periods following a fault condition, the flight protection response shall, at a minimum, autonomously configure the spacecraft to a safe, quiescent, ground commandable state, transmitting, at least an RF carrier downlink signal.
2. During critical mission activities (e.g., launch, orbit insertion), the flight fault protection response shall autonomously re-establish the needed spacecraft functionality to permit safe, reliable and timely completion of the mission critical activity.

2.6 Slosh Dynamics/ Mass Properties

1. The systems effects of propellant slosh dynamics and other sources of variability in spacecraft mass properties shall be accounted for, when applicable. In particular, the effects on stability, pointing accuracy, and fault protection, shall be addressed. Methods of positive mass property control (e.g. propulsion latch valves, trim orifices) shall be incorporated into the design to preclude unacceptable fluid migration or mass property change.

2.7 Information System Design and Margin - Data System and Telecommunications

1. The information system and telecommunication system design shall meet a default end-to-end downlink data quality average threshold bit-error rate (BER) $\leq 10^{-6}$ and an uplink threshold command BER $\leq 10^{-5}$ unless otherwise specified by the project.

Data System

1. To meet limited Deep Space Network tracking pass demands, the information system design shall consider significant use of data editing, data compression and improved data encoding techniques to meet downlink telemetry data requirements.
2. The information system design shall have bulk data storage capability to enable storage of time critical science data and/or engineering telemetry data during long non-track periods and accommodate for flight operational uncertainties caused by weather effects or ground tracking station problems.
3. The information system design shall use the *minimum* number of normal operations data modes as needed to meet the science/engineering requirements. Acceptable sub-optimum return shall be considered, particularly if cost/risk can be significantly reduced.
4. The information system design shall use the *minimum* number of normal operations telemetry formats to meet the mission science/engineering requirements. Acceptable sub-optimum return shall be considered, particularly if cost/risk can be significantly reduced.
5. The information system design shall have engineering emergency data modes and formats (measurements) for diagnostic use. A hierarchical measurement approach shall be used so that assessment of spacecraft health/safety can be rapidly attained.
6. The information system design shall provide adequate telemetry data to rapidly assess health status under *normal and faulted* operations. Special consideration shall be given to providing increased telemetry instrumentation for mission-unique or other sensitive functions.
7. The information system design shall provide sufficient telemetry data and sampling frequency, including any special diagnostics, to enable the flight team to perform anomaly determination, investigation/reconstruction, particularly for mission critical activities.

Telecommunications Design and Margin

1. The telecommunications system (end-to-end flight and ground telecom elements) shall be designed to meet the required information return, radio navigation and radio science requirements.
2. To reduce spacecraft mass and power demand, the Earth downlink shall be designed using the lowest practical power-gain product that meets the mission information return and quality requirements with appropriate margin, consistent with the mission/TMOD-approved Deep Space Network capabilities and tracking coverage.
3. Telecommunication equipment antennas and ancillary hardware shall be the minimum needed to meet the mission and system telecom requirements with acceptable risk and operating margin.
4. The spacecraft uplink shall be designed to accommodate an S-band or X-band carrier frequency. The spacecraft downlink shall accommodate S-, X- or Ka-Band carrier frequencies.
5. At implementation start (end of Phase B), nominal Deep Space link margins shall be at least 3 db. Deep Space links with extreme geometry conditions, surface-to-orbit links, or surface-to-surface links shall

consider 10 db or more margin, depending on the nature, complexity and scope of design uncertainties.

6. During implementation (Phase C/D), margins shall be probabilistically defined using appropriate statistical combinations of link parameter tolerances. Link margins shall be reported at PDR, CDR, ATLO start and Launch.
7. The telemetry system end-to-end design shall permit ground operators, early in the ground tracking pass, to determine rapidly and unambiguously the state of the spacecraft, particularly to determine if the spacecraft executed a fault protection response.
8. The design shall permit simultaneous command/telemetry capability using the same antenna or similar coverage antennas.

2.8 Thermal Design and Margin

Thermal Control Design Margin is the difference between the Flight-allowable (FA) Temperature range and the range between the worst-case hot and cold predicted temperatures. Worst case is that combination of realistic thermal extremes that produces the maximum hot and minimum cold predicted temperatures.

General

1. Temperature design shall be tailored to the specific applications of the mission with consideration for both equipment reliability and temperature/performance interactions.
2. Passive thermal design/approaches shall be used where practical. Active, complex thermal control design shall be avoided whenever possible.
3. The system thermal design shall control the subsystems to within the allowable flight temperature ranges.

Thermal Design Margin

1. The design shall have adequate thermal design margin to ensure no credible thermal threat to hardware when operating under normal conditions.
2. Bus electronics (at the mounting or thermal control surface for the specified assembly) and spacecraft mechanisms shall be qualified (by testing) for -35°C to +75°C OR FA temperature limits extended by -15°C and +20°C whichever is greater.
3. For credible abnormal conditions resulting from anomaly-induced power dissipation and/or off nominal sun attitude conditions, the thermal design shall maintain temperatures within FA limits extended by +/- 5°C (acceptance temperature range).
4. For non-credible, but plausible, conditions the thermal design shall maintain temperatures within FA limits extended by -15°C, and + 20°C (i.e. qual/protoflight levels)
5. The thermal design shall keep piece-part silicon junction temperatures less than 110°C (assuming a mounting surface temperature of 70°C) for the planned circuit design and packaging scheme. Higher junction temperatures may be considered where risk is shown to be acceptable or permitted by other technologies (e.g., GaAs).

6. Except for detectors, optics, and other instrument-unique hardware, the payload instrument electronics shall be designed to the spacecraft bus electronics requirements.
7. Optics, detectors and other unique hardware shall be designed for allowable flight temperature limits extended by -15°C and +20°C and margins may be tailored to specific application based on required operating temperature ranges of sensitive elements.
8. Thermal Cycling:
   - **Electronic** hardware design shall be capable of surviving power on-off temperature cycling and/or solar exposure cycling of three times the number of worst-case expected mission cycles with worst-case flight temperature excursions. Prior to having a mission estimate, the equivalent of 10,000 cycles with a 15°C delta T for new/inherited design hardware shall be used.
   - **Mechanical** hardware design thermal cycling profile shall be tailored for the specific application.
   - Flight hardware thermal cycling shall be minimized to preclude the risk of damage.

   Rationale: *Thermal cycling has been implicated as a major contributor to faults/problems.*

2.9 Propulsion Design and Margin

1. Propellant tank volume shall be sized to accommodate the nominal mission based on the required deterministic and statistical delta velocity needs (based on the total mass allocation), and appropriate ullage.
2. Statistical delta velocity estimates shall be based on 99% probability.
3. Propellant load estimates shall be based on specification minimum value Isp for engine/thruster and allocated spacecraft system mass.
4. Tanks shall meet the appropriate pressure vessel design and safety margin requirements under worst-case conditions.
5. Safe, reliable operation of propulsion subsystem components (e.g., valves, thrusters) shall be demonstrated by tests over a range of conditions that envelop flight operations expectations, with appropriate margins, (e.g., feed pressures, flow rates, mixture ratios, high voltages).
6. A component cycling usage margin of 50 percent or more beyond the worst case mission use shall be demonstrated based on the hardware heritage, prior mission use or qualification testing. Margin shall be reported at PDR, CDR, and ATLO start.
7. Hardware shall be thermally controlled to remain safely (>10 °C) above propellant freezing temperature whenever the hardware is in contact with propellant or propellant vapor.
8. Hardware that will come in contact with propellant vapor shall be thermally controlled over the entire mission to remain safely (>10 °C) above the temperature at which propellant condensation will occur.
9. Bi-propellant propulsion systems shall incorporate a passive means of ensuring that liquid fuel and oxidizer are prevented from mixing in the pressurization system or tanks.
10. Gas regulators (single or series redundant) shall be used to provide long-term isolation of pressurant from the propellant tank.

2.10 Prime Power Distribution/Switching and Margin

1. **Power System Grounding/ Fault Tolerance** - The prime power distribution hot and return lines shall be DC-isolated from spacecraft chassis by at least 2 K ohms.
   Rationale: *Ensure that a single fault short to spacecraft chassis anywhere in the distribution system between the power source, electronics and the user loads does not pose a catastrophic failure.*
2. **Load Removal** - Prime power on/off switching of electrical loads shall be done by "simultaneously" switching both hot and return sides.
   Rationale: *Ensure total load removal (no possible ground return sneak paths) in case of power-related faults.*
3. **Surge Control/ Load Removal** - Power interfaces shall be implemented with in-rush current surge suppression protection and with load removal capability to "clear" a load fault.
4. **Critical/ Non-Critical Load Selection** - A critical and non-critical prime power bus shall be considered. Hardware power bus assignment (critical or non-critical) shall be consistent with time critical mission load requirements and maintaining spacecraft safety and ground commandability.
   Rationale: *It is prudent to provide the maximum power margin practical post-power fault state for normal cruise operations.*

Power Converters

1. Subsystem off-the-shelf power converters shall be assessed to ensure compatibility with application and surrounding circuitry.
   Rationale: *Power Converter designs differ in their detailed signatures (ripple, spikes, transients, etc). Assessing sensitivity of user circuits to these details early can preclude costly problems later.*
2. Subsystem power converters shall be capable of operating via an externally supplied synch frequency signal or in a free-running mode, near the synch frequency.
   Rationale: *To minimize EMI effects.*

Interface Circuit Margins

1. At implementation phase start (phase C/D), there shall be 30 percent margin on the spare power switch and circuit count, including cabling and connector pins to accommodate late identified needs with minimum cost, schedule impact. Circuit count margin shall be reported at Project PDR, CDR, and ATLO.

2.11 Battery Energy Margin

1. At implementation phase start (phase C/D), the design shall have 40% or more energy margin (depending on new or inherited hardware/designs) assuming an allowable depth-of-discharge of 40% and current best estimate (CBE) of electrical load demand, including losses. Energy margin shall be managed and reported similar to power margin.
2. For solar array missions, battery capacity requirements shall account for nominal launch/ array deployment to cruise operational conditions, as well as appropriate margin for ground and/or space flight anomalies, and mission-critical modes.

2.12 Short Term Transient Energy Demands

1. The design shall consider capacitor bank energy storage to accommodate short-term large peak step loads, e.g., propulsion valve actuation.

2.13 PYRO Design and Firing Margins

1. The design shall have the capability to guarantee firing up to 6 NASA Standard Initiators (NSIs) simultaneously under worst-case conditions (temp, voltage, etc.).
2. Pyro circuits shall incorporate appropriate current limiting to control maximum circuit current flow.
3. The design of firing circuits shall avoid simultaneous arming of multiple functions without separate independent protection.
   Rationale: To avoid spurious unplanned pyro events caused by planned firings or other transient effects.
4. At implementation phase start (phase C/D) there shall be 30 percent margin on the spare pyro firing circuits, including cabling and connector pins. Circuit margin shall be reported at PDR, CDR, and ATLO start.
   Rationale: *To accommodate late identified needs with minimum cost, schedule impact*

2.14 Electrical Grounding and Interfacing

1. Grounding and Interfacing shall be implemented in the electrical and mechanical design (including packaging) to minimize EMI. The grounding and interfacing design shall:
   - Provide for an equipotential spacecraft, and "Faraday" cage where needed,
   - Provide low conducted and radiated emissions,
   - Provide high transient noise immunity on circuitry, and
   - Provide prevention or minimization of external and internal electrostatic discharge (ESD).

2. A static bleed resistive path using a 1 Megohm or greater resistor shall be provided *in each assembly* from circuit return to the assembly structure.
   Rationale: *to prevent charge buildup during periods when the unit is not mounted to the spacecraft.*
3. Structure or shields shall not be used for the primary circuit return path. Wires shall be used.
4. Each subsystem ground tree (i.e. power converter secondary) shall have a local single point DC ground to spacecraft chassis via the shortest practical wire length.
5. All non-coaxial interfaces shall use twisted-shielded wire pairs with shields grounded appropriately, unless other wire treatments can be used.
   Examples of other possible wire treatments are twisted pairs, triplets or no twisting at all depending on applications and the EMI threat.
6. High current, high di/dt and dv/dt interface wires shall be appropriately shielded/grounded. Furthermore, PYRO and power interfaces shall be physically separated from signal interfaces as much as practical, e.g., different routing and separate connectors.
7. Inductive loads (e.g. valve coils) shall be equipped with back-EMF transient suppression.
8. Space-exposed or "spacecraft-buried" ungrounded conductors shall be demonstrated to not pose an ESD disruption or damage threat. There shall be no ungrounded (floating) conductor > 15 cm in length.
9. Functions that pass through external spacecraft connectors (e.g. Umbilical, direct access) shall be protected in the event of inadvertent connection of any conductor to any other conductor and spacecraft chassis.
10. Electrical signals (e.g. data, timing, power, circuit returns) that use flexible cable or that cross mechanical interfaces shall be immune to transient signal interruption.
    Note: *The amount of transient protection depends on the function and sensitivity of the circuits involved.*
11. Power hot and return and chassis functions shall be adequately separated to preclude possibility of hot-to-return or hot-to-chassis shorts. Power connector pin assignments, cable routing, and electronic circuit layouts shall receive special engineering review/ oversight, particularly in designs where the prime power and circuit return and spacecraft chassis are in close proximity.
12. Electrical interfaces passing through cable cutter separation devices shall be dead-faced prior to actuation of the device, e.g., signal and power interfaces shall be unpowered.

2.15 Structural Design and Margin

1. Where cost effective (e.g., not driving mass to the extent that a new launch vehicle is required) the primary structure shall be designed with high safety factors (>2.0 ultimate).
   Rationale: *To preclude the need for structure verification static load testing.*
2. Static load testing shall be required for all primary structures, as a part of qualification, and to demonstrate margin.
3. Design shall consider using the most cost effective, lightweight materials to reduce mass as long as they are compatible with other design requirements, e.g., thermal, electrical and safety requirements.
4. Secondary structure design shall meet the load values from the mass/acceleration curve or test to the flight environments with appropriate margin.
5. The integrated design of structure, deployed appendages, and the attitude control response, shall preclude/minimize possible interactions caused by lower order modal frequencies.

2.16 Force/Torque Margin

1. Mission critical deployables design (e.g., solar arrays) shall demonstrate a margin of at least 100% under worst-case conditions, particularly cold, stiff cable bundles and considering vacuum versus air, and coefficient of friction effects.
2. Mission critical separations design (e.g., launch vehicle, probe release) shall demonstrate a margin of at least 100% under worst-case conditions.
3. Mission critical mechanisms and actuators design shall demonstrate at least 100% margin for the range of motion at the end-of-life conditions under worst-case conditions, including restart from any position within the range of motion.
4. "Helper" springs shall be used to assure first motion separation of surfaces where fraying/fretting is possible. "Helper" springs shall also be used to provide for guaranteeing latching of deployed elements.
5. Margins shall be reported at PDR, CDR, and ATLO.

2.17 Radiation Design Margin (RDM)

Note: *RDM is a design factor to be applied in the design specification of electronic parts and part application design.* ***It is not a reserve****, or other resource which can be used up during the design*

RDM is defined: $\frac{\text{electronic part capability}}{\text{electronic part expected local environment}}$.

1. RDM shall be calculated based on the Current Best Estimate (CBE) plus ***reasonable*** margin to accommodate uncertainties for space environment, transport modeling, and part capability.

Notes:

- *Shielding to an RDM of 2 (traditional goal value) is required at the end of the nominal mission unless the project can demonstrate acceptable risk with lower margin.*
- *Circuit design margins are currently calculated including the combined effects of radiation, temperature, aging, voltage variations, etc. Voltage and temperature are major contributors.*
- *Voltage and temperature effects may be traded for radiation effects at some risk.*
- *A higher chance of degradation at/near the end of the mission may be accepted, provided that mission success is not dependent on at/near end-of-mission events.*
- *Where spot shielding of a component is to be applied, an RDM of 3 is required to account for greater modeling uncertainties.*

2.18 Graceful Degradation

1. The design robustness shall include consideration of:

- Inadvertent operation outside expected flight environments, e.g., temperatures, radiation dose
- Shortfalls in performance, e.g., RF power output, antenna gain
- Fault propagation due to collocation of components, e.g., thrusters, adjacent redundant electronic components on the same chip.

Rationale: *To reduce possibility of catastrophic mission loss or major mission degradation.*

2.19 Configuration Design and Fields-of-View (FOV) Interactions

1. The configuration design shall provide an appropriate amount of additional clearance beyond nominal specified FOVs to preclude /minimize obscuration effects, (e.g. to sensors, antennas and thrusters), caused by structural elements, blankets, booms, etc.
2. Straylight input shall be considered and effects precluded /minimized, particularly for attitude control celestial reference sensors and science imaging instruments (e.g. visible, IR and UV spectral regions).
3. Thruster or external venting plume impingements shall be precluded/ minimized.
4. RF antenna pattern distortion effects shall be precluded/ minimized.

2.20 Interface Commonality

1. The system design shall use a common electrical interface approach and circuits to reduce interface designs and protocols.
2. The system design shall minimize the number and type of interface approaches/circuits used.
3. The system design shall consider the use of proven reliable interface types where fault issues, etc. have already been addressed, e.g., 1553 data bus or other avionics standards.

2.21 Reliability Analyses/ Design Confidence

1. Mission/System-level fault tree analyses (FTAs) shall be performed and maintained/updated throughout the project life cycle. The most recent FTA shall be presented at PMSR, Project PDR, CDR and pre-ship reviews.
2. The Design shall be assessed for robustness through a program of analyses tailored from the Reliability Analyses Handbook (JPL D-5703) or Contractor/Partners equivalent, including Part Parameter Data from available databases, and Derating Guidelines (JPL D-8545). Analysis/test types to be performed shall include:
   Worst-case circuit analysis, Voltage-Temperature- Frequency margin testing (where it is feasible and prudent) to demonstrate performance margin.
   Failure mode effects functional analysis (FMEA) at the system/subsystem functional block diagram and interface levels - identifies potential critical single failure points.
   System interface circuit, functional, and fault analyses (mechanical, thermal, etc.) - demonstrate that faults in one subsystem/system will not propagate or functionally degrade other subsystems.
   (Failure Modes Effects/ Criticality Analyses (FMECAs) are generally applied to electronics and electronic functional interfaces, and subsystem mechanical Fault Tree Analyses (FTAs) to devices and mechanisms).
   Parts stress analyses - verify margins.

2.22 Electronic Parts Usage

General

1. Appropriate derating of parts shall be incorporated in electronics design.
2. The availability and cost/risk effectiveness of grade-one parts shall be considered before COTS parts become the design baseline.
3. An early design parts list review shall be performed against documented requirements to:
   - Identify long-lead time parts.
   - Assess radiation dose, latch up and Single Event Effects (SEE) capability/compatibility.
   - Minimize the number of different part types.
   - Provide parts vendor assessment information.
   - Assure all known parts issues are identified and closed early.
   - Benefit from Parts Engineering/independent assessments and knowledge from other missions.
   - Provide data to project risk database.
   - Cost-effective match between design and parts capabilities.
4. The root cause of electronic parts failures shall be determined.
   Rationale: *Avoids repeating same or related failure, and develops effective and efficient corrective action that addresses underlying cause.*

ASICs and FPGAs

5. Mixed signal (digital and analog) ASICs shall be considered to meet packaging and power constraints/objectives.
6. ASIC design shall develop behavioral and hardware description models to capture implementation of system design specifications and evaluate performance.
7. Test vectors shall be developed and simulations performed to demonstrate the hardware description model design matches behavioral model, the gate level model matches the behavioral model and fault containment is understood.
8. Functional tests shall be performed with simultaneous digital, analog and mixed signal circuitry to assess interactions, as well as, separate tests on each portion of the ASIC.
9. Analog, digital and mixed signal ASICs shall be modeled or simulated and compared with test data.
10. Analog and digital ASICs shall be wafer-probed at room temperature and at maximum rated operating temperature.
11. Margins shall be maintained for the allocation of gates to implement an ASIC application. Depending on the complexity and maturity of the design, margins shall be at least 15 percent at CDR.

2.23 FPGA/ ASIC Transient Operations at Power Turn-On/ Turn Off

1. Precautions (e.g. time-out) shall be taken to prevent adverse effects due to the unpredictable logic states of FPGAs and ASICs, which can occur at power-on and power-off.
   Rationale: *During power turn-on or turn-off, FPGAs / ASICs may be in unpredictable logic states for several 10's of milliseconds.*

2.24 Synchronous/ Asynchronous Digital Design

1. Synchronous design shall be used for digital logic to guarantee the sequence of logical decisions and the validity of data transfer.
2. The synchronous design of ASIC or FPGA shall be verified, as a minimum, by post-route timing analyses using a place and route tool and test vector simulation with timing checkers performed at the primitive level. Timing of boundary conditions (pin-outs) shall be constrained both for place, route, and test vector simulation.
   Note: *Asynchronous design may be used if techniques are employed and demonstrated to provide guarantees for sequence verification and validation to the same confidence level as used for a synchronous design.*

2.25 Systems Safety

1. System Safety analyses, inspections and tests, and required reports, shall be performed according to the guidelines and requirements of JPL Standard for System Safety (D-560). These include:
   - A preliminary hazard analysis- in support of preparation of System Safety Plan
   - A Safety Compliance Data Package
   - Safety tests and/or inspections, and Facility and operational Safety Surveys

2.26 Environment Compatibility Verification

1. Environmental design assessments and verification tests shall be performed to verify the design against the specified environment. These shall be performed at the unit, and system level, considering the requirements and guidelines of JPL D-14040, "Process and Technical Guidelines for Spacecraft Hardware Project-Specific Environmental Assurance". Such analyses and tests may include:
   - Analyses - Single Event Effects (SEE), micrometeoroid, pressure profile, magnetic fields, etc.
   - Unit-level Qual random vibration, pyro, thermal, EMC, and Acceptance random vibration and thermal
   - System-level/ Protoflight random vibration and/or acoustic, pyro shock, thermal vacuum, EMC

2.27 Electronics Minimum Operating Time

1. A minimum power-on operating time shall be established for all electronics as follows:
   - Unit Level prior to spacecraft integration**:** each electronic assembly, including each side of a block-redundant element, shall have at least 200 hours operating time.
   - System Level prior to launch: each single-string electronic assembly shall have 1000 hours operating time. Each side of a block-redundant element shall have at least 500 hours operating time with a goal of 1000 hours.

2.28 Quality Assurance Verification and Validation

1. JPL source QA provisions shall be provided for critical processes/products and strategically applied to high-risk suppliers.
2. Analyses, inspections, and/or tests shall be performed to ensure that the as-built product is consistent with the as-designed Baseline Configuration.
3. Quality assurance provisions, as defined in the Project QA Plan, shall be implemented throughout the ATLO process. Such provisions may include:
   - Work proactively in the safety and contamination control activity to ensure hardware integrity.
   - Provide configuration support for test and flight software.
   - Assure that project documentation requirements are met.
   - Conduct a physical verification of all hardware - to ensure that it meets the workmanship, CM and other project requirements.
   - Witness Critical operations.
   - Maintain spacecraft/instrument configuration log.
   - Remain an integral part of the SRCR/HRCR process.

2.29 High Voltage Power Supply Controls

1. High voltage power supplies shall have at least two independent, separate controls to activate/deactivate high voltage to assure that no single fault/command can result in high voltage state, which may result in risk to personnel or hardware, or be a mission safety hazard.

Section 3 - Flight Operations Principles

3.1 Operability

1. The flight systems and flight operations design shall be developed concurrently to enable cost-effective end-to-end operations.
2. The flight systems shall consider methods to reduce operational complexity and interdependencies (e.g. require less calibrations, provide more on-board closed-loop control, provide robust technical margins, provide more autonomy).
3. Operability design trades conducted and attributes incorporated shall be identified at the flight systems PMSR, Project PDR and CDR.

3.2 Flight Operation Sequences

1. Flight sequences shall operate the spacecraft consistent with flight rules provided by the developers and within environments and functional regimes experienced during development testing. Any planned operation beyond that ground tested shall be tested prior to flight use to demonstrate safe, reliable functionality and acceptable margin *OR* shall be approved by the Project Manager.
2. All flight sequences shall have been tested on a high fidelity flight-like system test bed and all anomalies dispositioned prior to sequence uplink transmission.
3. Standardized sequencing techniques shall be used for repetitive sequencing activities to reduce cost and risk.
4. For mission time-critical sequences (e.g., launch, orbit insertion), the driving design requirement shall be safety and reliability, even at the expense of reduced performance.
5. After initiation, mission time-critical operations shall *not* require "ground-in-the-loop" commanding to enable successful operation/completion.
6. The launch sequence and other mission critical sequences shall be test-verified on the spacecraft before launch under nominal and faulted conditions using the final load flight software. If resources or other factors do not permit testing of critical mission sequences, the system test bed may be used for verification.
7. Completion of the launch sequence shall leave the spacecraft in a ground-commandable, safe state requiring no "immediate" time-critical ground commanding to assure health/safety.
8. Flight sequences to be used within the first 30 days after launch (e.g., Trajectory Correction Maneuver) shall be test-verified on the spacecraft prior to launch.
9. Flight software loads/updates and sequence memory loads, particularly for those affecting mission critical capability, shall be verified by a memory readout or checksum readout. Depending on the application and mission/system consequence, single or multiple readouts shall be considered.

3.3 First Time In-flight Events

1. First in-flight use of functionality, particularly for mission critical or irreversible events, shall receive special development attention (e.g., analyzing what ifs, reviewing Red Flag PFRs, significant PFRs /ISAs, identifying need for additional testing, identifying need for contingency plans) during the sequence development process to assure safe, reliable flight operation.

3.4 System Test Bed - Spacecraft Fidelity

1. After launch, the ground system test bed configuration/state shall be maintained as close as practical to the flight spacecraft state, particularly the flight software code, parameters, counters, etc., to minimize test initialization and run times, and to provide high confidence in the test bed results.

3.5 Contingency Plans

1. For at least mission critical and first time in-flight events, contingency plans shall be developed to minimize the threat to health/safety in case of unexpected/improper spacecraft response.
2. All contingency commands shall be system test bed-verified prior to transmission to the spacecraft. Additionally, Launch related contingency plans should be test-verified on the spacecraft

3.6 Operating Margins

1. Adequate operating margins (e.g. memory, timing, power) shall be maintained for all stored sequence controlled and real-time flight activities to maximize the prospects for safe, reliable operation.

3.7 Telemetry Predicts and Alarm Limits

1. Subsystem telemetry measurement predictions and alarm limits shall be developed and in-place prior to planned spacecraft operations to provide rapid assessment of operational performance and provide an early alert of potential spacecraft problems.

3.8 Powering off the Spacecraft Downlink

1. After in-flight turn on, the spacecraft downlink RF transmitter hardware (e.g., exciters, power amp) shall not be turned off during nominal flight operations. The transmitter shall remain powered during the entire mission unless momentarily power cycled via system autonomous fault protection responses.

3.9 Maintaining Spacecraft Health/Safety

1. Spacecraft operations shall be consistent with maintaining health/safety. Hence, unnecessary risk-taking shall be avoided. If health/safety flight rule violations are necessary to implement activities, violations shall be approved by the Project Manager.

3.10 Fault Protection (F/P) Value Limit Strategy

1. Spacecraft Autonomous fault protection enable/ disable strategy, threshold trigger values, and persistence values shall be established considering mission phase applicability and operational activity. The enable/disable, trigger, and persistence values shall be selected to ensure safety but not "hair triggered" to cause inadvertent F/P entry/ execution.

3.11 Spacecraft Characterization and Evaluation

1. The flight operations team shall consider early demonstration of spacecraft functional capabilities prior to actual mission need to characterize and evaluate the spacecraft and ground system end-to-end operation. Early characterization/evaluation enables the Project to identify flight/ground system shortfalls and make changes safely and reliably with minimal threat to the mission.

3.12 Power Cycling and Prime/ Redundant Hardware Usage

1. Power cycling of mission-critical hardware shall be avoided.
2. Prime selected hardware elements shall remain in use for all operations.
3. Swapping to redundant hardware elements shall be limited to fault recovery actions to assure health/ safety.
4. Simultaneous use of selected prime and redundant hardware to enhance reliability/performance for accomplishing mission critical activities shall be considered only after careful study, and shall be approved at the Mission Event Readiness Review.

3.13 Redundant Ground Coverage

1. Redundant ground coverage (e.g., site or antenna at same site) shall be planned during mission critical operations to guarantee real-time performance visibility and enable ground contingency commanding, if necessary.

Appendix A: Software Principles

A.1 Introduction

Recent changes in JPL's external environment have resulted in the establishment of many small flight projects operating under tight budgets and schedules. Indeed, the Laboratory has set a goal of significant reductions in both cycle time and cost, while maintaining the quality of the products we deliver. This places a premium on concurrent development and careful process tailoring within the context of proactive risk management.

Traditionally software development has been addressed well into the project life cycle as a third- or fourth-level design consideration. However, tighter development schedules, plus the growing pervasiveness of software throughout the mission system has made it mandatory to define, design, and implement flight software in a more disciplined manner. Mission success (including personnel and equipment safety) is of paramount importance; risk, cost, and schedule are managed accordingly.

Although applicable to other domains, mission-critical software is the primary target of the software development principles contained in this document. Mission-critical software is identified by each project and typically includes flight software as well as software used in the uplink, downlink, and navigation processes. (See Section 1.0 of D-15378 for guidance in identifying mission-critical software.) In addition to software, these principles also apply to the development of mission-critical firmware, through completion of testing in a simulated hardware environment. "Software development", as used here applies to both pre-delivery and post-delivery development activity; the latter is often called "maintenance".

The principles are intended to foster the needed discipline by documenting good practice within the JPL environment. They elaborate the broader principles contained in this document and at the same time provide additional guidance on meeting the requirements documented in The Software Development Process Description, D-15378. The principles come from both JPL's own experience (lessons learned) and from the literature on software engineering and management (good practice elsewhere, not yet widespread at JPL).

Process oversights are the root cause of many of JPL's recent mission failures. Moreover, inadequate planning and ineffective implementation are common contributors to schedule and cost overruns. Thus, this initial version of software development principles emphasizes the requisites of good software process in the JPL environment. The last section focuses on detailed design and development practices for flight software. It is planned to expand the treatment of detailed design principles in a subsequent version, paying particular attention to the differing needs of flight software, instrument software, telecommunications, telemetry and command, navigation, and

science data processing.

It is intended that these software principles be used in the same fashion as the principles in the body of D-17868, namely:

- Software development — both in-house development and subcontracted/partnered development — should comply with each applicable principle unless there are good reasons to take exception.
- Compliance with each General Principle (or the rationale for not complying) is documented in the Project Implementation Plan, which summarizes compliance details documented in the project Software Development Plan. Compliance will be verified during reviews.

These general principles are loosely organized around life cycle activities. The statement of a principle is bold numbered; supporting text appears as italic font. Supporting text is not part of a principle; it is meant to illuminate the intent of the principle. Principles are numbered within each topic to facilitate use. In addition to the published sources listed in Section 2.0 of this appendix, these software principles drew upon the expertise and experience of a broad spectrum of JPLers representing practitioners, and both line and project management.

Note that it is not the intent of this document to prescribe or preclude any software development life cycle.

A.2 Published Sources

1. N. Brown, ed., The Program Manager's Guide to Software Acquisition Best Practices, version 2.0, Department of Defense (1996).

2. A. M. Davis, 201 Principles of Software Development, McGraw-Hill (1995).

3. J. Hihn and H. Habib-agahi, Flight Software Cost Risk: Analysis and Recommendations, JPL D-18409 (January 2000).

4. M. Landano and J. Rose, Design, Verification/Validation and Operations Principles for Flight Systems, JPL D-17868 (June 1999).

5. C. Lin, Project Close-Out Plan, Alaska SAR Facility Development Project, JPL D-18032 (August 1999).

6. C. Lin and H. Kea, GSFC/JPL Quality Mission Software Workshop Report Held on August 17-19, 1999 in Annapolis, MD (October 1999).

7. NASA Course on Software Acquisition Management (September 1997).

8. Report on the Loss of Mars Polar Lander and Deep Space 2 Missions, JPL D-18709 (March 2000).

9. S. McConnell, Rapid Development, Microsoft Press (1996).

10. Software Development Process Description [SDPD], JPL D-15378, Rev. D (1999).

11. R. Stutzke, in Proc. 7th European Software Control and Measurement Conference (1996).

12. R. Thayer, ed., Software Engineering Project Management, IEEE (1997).

A.3 General Principles for Software Development

The general principles of software development applicable to JPL flight projects are organized into the following topics:

1. System Definition/System Engineering
2. Planning and Monitoring
3. Cost Estimation
4. Software Risk Management
5. Organization and Staffing
6. Design and Implementation
7. Integration and Test
8. Configuration Management
9. Software Acquisition
10. Product and Process Verification

A.3.1 System Definition/System Engineering

1. All requirements shall be organized in a comprehensive framework, formally documented, and traceable to higher-level requirements where possible.
A summary of the verification effort should be reported at all major reviews. Some milestones where verification reports should be considered are:
- *Demonstrate that the software architecture will address the top-level requirements*
- *Verify that the design covers all the requirements*
- *Ensure that the test plan verifies all requirements*
- *Review the test results to assure compliance with requirements.*

2. Requirements shall be prioritized.
Requirements are prioritized to facilitate implementation planning and possible changes in scope and budget. In most cases, a categorization of requirements will suffice -- exhaustive rank ordering is not needed. System level requirements allocated to software should be prioritized to separate mandatory from desirable features and should be labeled to indicate the time phasing of planned implementation.

3. All internal and external interfaces shall be defined and documented.
Interfaces between flight hardware and flight software should normally be the responsibility of system or subsystem engineering.

4. Functional requirements shall be validated against a concept of operations early.

5. Prior to implementation, the end user shall have the opportunity to review the functional requirements and approve them, as appropriate.

6. Hardware, software, and operations engineers shall jointly perform hardware/software trade-offs and develop an integrated system and software architecture that addresses the mission requirements and operations concept.
7. The mission software architecture shall be developed and documented prior to approval of the Project Implementation Plan (PIP). The architectural design shall be subsequently updated and reviewed at all significant reviews.
8. A software design shall be simple and modular to facilitate development, debugging and testing, future modifications, and the rapid understanding of its logical structure. Important concepts to apply include:
   - Cohesion and coupling
   - Information hiding (encapsulation)
   - Standardization of interfaces and data structures
   - Ease of testing.

   A good modular design exhibits strong cohesion or affinity among the functions assigned to each module and minimal coupling or interfaces with other modules. Encapsulation within the module of information needed by that module alone further promotes this concept.
   When the design process of one element creates requirements for another element, the schedule, WBS, and budget should reflect these interdependencies.
9. The software architectural design and development plan shall be engineered to accommodate probable requirements change.
10. Proposed requirements changes shall be documented and assessed for cost, schedule, and technical impact before being accepted as a delivery commitment.
11. There shall be a formal review of software requirements prior to implementation.
    Multiple reviews may be required for iterative development.
12. System engineers shall review software integration and test results to identify and document unforeseen but allowable system idiosyncrasies.
13. The developer of a multi-mission system shall provide a potential project user with the following items:
    - An architectural design document
    - As-built capabilities and design documentation
    - Test plans, procedures, cases, and results
    - Development and testing tools used for the pertinent version, plus user's guides for the delivered tools
    - An adapter's guide
    - Training to facilitate project adaptation.

A.3.2 Planning and Monitoring

1. The software development plan shall be written in a fashion to communicate a shared vision, define goals, and assign responsibilities to participants.

   This shared vision should be refined and articulated throughout development.
2. The software development plan shall be developed in concert with the Project Implementation Plan (PIP), address topics in The JPL Software Development Process Description and NPG 7120.5A insofar as applicable, and shall comply with JPL software quality assurance policy.

A list of topics to pay particular attention to follows:

- *Integrated schedule*
- *Reuse strategy, if applicable*
- *Sub-plans (as needed) for*
  - Incremental builds
  - Integration & test
  - Configuration management
  - Risk assessment and management
- *Verification of procured items*
- *Progress/product metrics*
- *Change control for requirements and design documentation*
- *Identification of quality records*
- *Peer reviews*
- *Design, implementation, and operation of the development and test environment*
- *Testing philosophy*
- *Problem reporting*
- *NASA Independent Verification and Validation (IV & V).*

3. The development plan shall employ an incremental or iterative approach to implementing and testing system components.

4. Implementation shall be planned at a level of detail that facilitates tracking the progress of individual developers or small teams.

5. Interdependencies among major activities shall be negotiated, captured, and maintained in a network schedule or equivalent, with the critical path indicated.
Schedule margin should be consistent with budget reserve.
The ground support equipment and simulation software schedules should be responsive to the flight software schedule.

6. The development plan shall provide for design, early implementation, and validation of both the development environment and the required test facilities, with particular attention to the needs of physically distributed development.

7. Concurrent development of interfacing hardware and software shall be jointly planned and shall be coordinated via integrated hardware/software peer reviews and the joint preparation of integrated test plans.
To reduce costly rework and schedule delays caused by independent development of interfacing hardware and software, close cooperation is required by the separate development teams. Joint planning and joint peer reviews nurture cooperation.

8. Planning shall provide for training developers, testers, users, operators, and maintainers.
In addition, developers should be briefed on system structure and mission goals.

9. Reviews shall be identified in the software development plan. Reviews may be combined, as appropriate.
Software reviews of interest to the project as a whole shall be incorporated into the project review plan. The following topics should be addressed in designing a project review scheme:

- *Commitment to a proposal or work package*
- *Completion of user requirements*
- *Completion of development plan and risk management plan*
- *Architectural design, addressing interfaces and interactions between modules*
- *Inheritance review that addresses legacy code, reusable components, and COTS; support by original developers and cost estimates are key topics.*
- *Technology readiness*
- *Completion of system/subsystem requirements and design (as needed)*
- *Completion of software requirements and design (iterative development may necessitate multiple reviews); inheritance should be addressed explicitly.*
- *Test architecture and test plan, including design of testbeds, simulators, and models*

- *Test readiness*
- *Functional validation or pre-acceptance test*
- *Requirements for Software Review/Certification Record (SR/CR)*
- *Peer reviews.*

10. Review of a software design and its implementation in code shall include technical experts external to the project, hardware engineers (e.g., product integrity engineers) who understand the design and function of all interfacing hardware components, and representatives from the operations team.

11. If significant software inheritance is planned, a review shall be held prior to project PDR.
A software inheritance review should be part of the project inheritance review. Its purpose is to establish feasibility and risk, and to estimate the additional development effort required. The approach to inheritance should be reviewed at PDR and CDR as well.
Note: *A key risk factor in inheritance is the accessibility to the individuals who developed the software that is a candidate for inheritance.*

12. Peer reviews shall be applied to requirements, designs, code, test plans, test results, and documentation.
Stable membership of peer review teams is recommended.
The effectiveness of reviews may be enhanced by using checklists of common errors and critical issues.

13. Development progress and product quality shall be tracked by metrics, tailored to project needs.
Recommended metrics are:

- *Resources and Cost (e.g., planned vs. actual effort)*
- *Schedule and Progress (e.g., for each build, percent completion of requirements, design, code, and test)*
- *Product Quality (e.g., a plot of PFRs opened vs. closed; percent of code tested; rework effort)*
- *Growth and Stability (e.g., source lines of code; requirements volatility)*
- *Processor Capability (e.g., time history of identified margins).*

14. Anomalies, change requests, and liens shall be documented, dispositioned, and tracked.

15. Adherence to the development plan and the test plan shall be reviewed periodically, and these plans shall be revised as appropriate.

16. Transition to operations and maintenance shall be guided by a plan addressing topics in D-15378.
Pertinent topics include the following:

- *A user's guide and operator's manual*
- *Detailed description of operational idiosyncrasies, known problems, and requested changes; resolution of problems and implementation of changes should be prioritized*
- *A complete set of test cases and test reports*
- *Identification of tools and associated databases used in design, coding, integration and test, configuration management, installation, and the tracking of defects/changes*
- *Identification of reports, studies, and data pertinent to improving product performance*
- *As-built design documentation at a level of detail needed to familiarize the maintainer with the software structure and function; design documentation should include coding standards and design rules.*
- *Identification of other developer responsibilities necessary to support transition to maintenance.*

17. A close-out plan shall provide for documentation and publication of lessons learned (with particular attention to updating software principles) as required by NASA NPG 7120.5A.

A.3.3 Cost Estimation

1. All cost estimates that are the basis for commitments shall be based on a bottoms-up estimate derived from an architectural design and an operations concept.
Cost estimates should address:
- *Documentation*
- *Maintenance/upgrade of development tools*
- *Software support to the modeling and simulation of hardware systems*
- *Development of simulated input data in the absence of actual input data sets*
- *Realistic support to project integration and test/ATLO*
- *Training the staff in the use of development techniques/tools and familiarizing the staff with the mission systems*
- *Transition to operations and maintenance*
- *Personnel turnover.*

A top-down estimate based on analogies to completed development may be useful as a cross-check on a bottoms-up estimate. To prepare for changes in scope and budget, it is recommended that cost be mapped to a work breakdown structure that includes both development tasks and development support functions, such as configuration management and integration & test.

2. Use of inherited software, multi-mission software, COTS, and public domain software modules or tools that have not been characterized, or whose internal functions must be modified, shall be budgeted and scheduled, based on an analysis of the required adaptation effort.

3. A cost estimate shall contain a funded margin that is based on a comprehensive risk management plan and staffing profile. The process for deployment of budget and schedule reserves shall be documented.

4. Estimates of the cost-to-complete and time-to-complete the development shall be prepared periodically and examined at milestone reviews.
Quarterly updates are recommended for cost-to-complete and time-to-complete.

A.3.4 Software Risk Management

1. A project shall prepare a software risk management plan that identifies and prioritizes risk items (risk list), trigger events, descoping, and other risk mitigation options. The software risk management plan shall be prepared prior to the start of design, be updated periodically, and be reviewed at Project PDR and all other significant reviews.
   A software risk management plan can be a section in the software development plan. An important adjunct of this plan is a Risk List that is used to track and disposition development risk items. Common risk items are:
   - *Requirements that are to-be-supplied or need additional definition*
   - *Unverified assumptions and unknown/indeterminate performance parameters*
   - *Interfaces that have not been verified*
   - *Tight margins*
   - *Critical schedule interdependencies, especially for work done out-of-house*
   - *Performance of COTS software*
   - *Testbed availability.*

2. Software safety/hazard analysis shall be completed and integrated into the project's risk assessment early in the life cycle.
3. The development plan shall provide for early validation of interfaces, high-risk algorithms, and COTS. Insofar as practical, software shall be developed in risk order, with initial attention to the highest risk items.
   It is recommended that prototyping be used during Phase B of a project to 1) get an early start on the design of difficult functions, and 2) establish team-to-team interfaces and stabilize the development process. Early implementation of the core architectural elements (software principle 3.6.10) addresses interface risks, which comprise a category of risks common to most development. Software principle 3.6.10 reinforces this idea by requiring early resolution of the interfaces inherent in the core elements. The intent is to get the core software running in parallel with prototyping and implementation of other high-risk software elements.

A.3.5 Organization and Staffing

1. To ensure project-wide coordination of software tasks, each project shall have an experienced software manager with overall responsibility for the development and integration of flight and ground software systems.
2. To ensure project-wide coordination of software efforts with other project elements, each project shall have an experienced software system engineer.
   This person's responsibilities should include:
   - *Negotiation of requirements*
   - *Review of requirements, designs, and plans*
   - *Control of the software design*
   - *Negotiation of interface agreements*
3. The staffing plan shall provide for filling key software roles early, provide for staffing through ATLO and mission operations, and plan for an orderly transition of staff during close-out.
4. Early in the design phase, a project shall establish the role of software system architect, who has the authority for developing and communicating a vision of the structure and function of the mission software system and its relationship to hardware subsystems.
   The software architect is responsible for formulating the design and implementation philosophy, as described in the development plan, and for communicating this philosophy to developers, system engineers, and integration and test personnel. The software architect is a role distinct from the software manager identified in software principle 3.5.1, and from the software system engineer identified in 3.5.2. Nevertheless, in some cases one individual may occupy two or all of these roles.
5. The software architecture shall be documented before staffing up the implementation team.
   Preparing the software architecture before staffing up the implementation team is meant to ensure that software designers and programmers will be able to apply their effort effectively.
6. If it is likely that software will be used to solve interface problems late in development, the staffing plan and budget shall provide for retention of a cadre of experienced development and test personnel through ATLO.

A.3.6 Design and Implementation

1. The design documentation shall include an explicit identification of software configuration items and their relationships, and shall include a narrative that documents performance, quality of service, assumptions, and constraints.
2. The software design shall incorporate the appropriate level of functional flow-charting (or equivalent overview of the software design) prior to the start of code development.
3. The software design shall be verified by a trace to software and mission requirements. To ensure that each software requirement is captured, justified, and properly interpreted, a two-way trace of system requirements down and software requirements up, shall be performed.

4. Software logic design and its implementation in code shall be based on:
- Diagrams that depict flow of control, state transitions, or equivalent graphics that facilitate comprehensive evaluation of the execution paths
- An analysis of possible software failures
- Explicit consideration of off-nominal behavior and possible failure of interfacing hardware components.

5. Design reviews shall explore potential performance issues (e.g. depth of queues, size of arrays, task starvation).

6. Parameter values in flight software input and output data files shall:
- Specify both a nominal value and an allowable range appropriate for trapping errors
- Document and verify the derivation or origin of both the nominal value and the allowable range of values.

7. The software logic shall verify that each parameter value in a database or data file that is either input to or output from the software falls within an allowable range and shall provide for fault correction and recovery in the event that allowable ranges are violated.

8. Prior to implementation, the design of an Application Specific Integrated Circuit (ASIC), a Field Programmable Gate Array (FPGA), or any hardware containing embedded code, shall be reviewed jointly by the hardware and system designer(s) and by those responsible for the design of interfacing software.
Design complexity can have multiple downstream development impacts. These impacts should be identified early in the design process to ensure that the agreed upon design has the required flexibility to handle probable changes in higher level design requirements while permitting a joint hardware-software development that 1) falls within the combined budget and schedule constraints and 2) provides for adequate design, integration, and test of all components.

9. Margins shall be established early for critical performance parameters such as CPU speed, control cycle rates, interrupt rates and durations, communications bandwidth, RAM, and EPROM sizes. Margins shall be tracked continually and shall be re-examined in conjunction with significant design changes.

10. New technologies, tools, and architectural approaches to be employed shall undergo a formal technology readiness review and shall be assessed as potential risk items.

11. The core software architectural elements shall be implemented early.
Identification and resolution of fundamental integration problems is greatly facilitated by early implementation of core elements that permit end-to-end execution of the code. Code review for critical architectural components should reduce initial integration problems.

12. A software development shall employ institutionally supported development tools.

13. Design of the development environment shall address protection against unauthorized access, virus protection and removal, periodic back-ups, and disaster recovery plans and procedures
Periodic off-site back-ups are recommended.

14. A software development shall establish and monitor the use of documented design rules and coding standards.

15. The delivered software product shall comprise both code and as-built documentation.

A.3.7 Integration and Test

1. The design shall enable software testing at unit, module, subsystem testbed, and system testbed levels to incrementally verify functionality/operability.
2. The overall test activities shall be focused on critical functional areas.
   Monitoring of test coverage is strongly recommended. All paths of importance to flight operations should be identified and tested -- corresponding to both nominal and off-nominal use.
3. Formulation of a written integration and test plan shall be done concurrently with the development of software requirements and shall include a plan for the validation of documentation for end users. The test plan and status shall be updated and reviewed at PDR, CDR, and all other significant reviews.
4. The integration and test plan shall comply with the requirements in JPL D-15378.
   A list of topics to pay particular attention to follows:
   - *The multiple levels and scope of testing to be performed*
   - *Test scenarios, test cases, and test tools to be acquired or developed in order to realize the required code coverage*
   - *Use of automated testing to expand test coverage*
   - *Maintenance of a regression test suite*
   - *Use of the institutional problem reporting system to give the project visibility into test status (use of a separate anomaly tracking tool by the development team is not precluded).*

   For real-time systems such as flight software, it is important to test initial condition states -- e.g. boots/resets of the target system and all interfacing systems -- to conduct multi-day tests, and to exercise redundancy continually.
5. Independent testers shall be used to verify requirements compliance for mission- critical software.
   Use of testers who are separate from the implementation team should be considered for verification of all flight software. Participation of operations engineers in pre-delivery testing is highly recommended.
6. Test planning shall include detailed plans to verify the correctness of the software during transition from one mission phase to another (e.g. from cruise to planetary encounter, from cruise to entry/descent/landing (EDL), or from EDL to landed).
7. Test planning shall be guided by a failure modes and effects analysis (FMEA) and/or fault tree analysis (FTA) that considers 1) possible software failure modes and 2) both failure and off-nominal behavior of interfacing hardware components.
   The test plan should specify the extent of exception testing, error injection testing, and transient testing.
8. In systems that have redundant processing strings, the testing shall be varied to exercise all strings adequately -- e.g., don't always run on string A.
9. Test planning and the design of test cases and test procedures shall be based on the premise that the software contains serious errors that must be detected via thorough identification of off-nominal, implausible, and otherwise unexpected conditions arising from:
   - Defective software logic design
   - Incorrect initialization of parameter values
   - Erroneous parameter values in data input files
   - Hardware failures and transient or anomalous hardware behavior, and unexpected hardware-software

interactions

- Processor resets.

The test team should be energetic, creative, and persistent in their efforts to "break the software".

10. Regression testing shall be conducted systematically to verify correctness of both changes to software logic and additions/deletions/changes to parameters in data input files.

11. A test shall be repeated from the beginning if the test is aborted or flawed.

12. The Preliminary Mission System Review (PMSR), PDR, and CDR shall include a description of the development environments, test facilities, and simulation capabilities to be employed. The capabilities provided by each testbed shall be presented, and the capabilities lacking from the testbed set shall also be identified.

13. Software traceability from final system test back to mission requirements shall be demonstrated, with each user requirement traceable to one or more test cases.

14. Unit testing shall be required for inclusion in a build. The systems engineer or his delegate shall define pass/fail criteria for unit testing. Unit testing shall, at a minimum, test against the full operational ranges of parameters.

15. Written defect tracking shall begin at the earliest practical time. Defect tracking shall include documentation of problems with the development environment.
It is often helpful to track defects found in requirements and design.

16. The defect tracking system shall identify the release or version where the defect was found and the release where it was remedied. The defect tracking system shall also reference any documented problems related to the defect (e.g. ISA, PFR, PIR).

17. Pre-delivery tests shall be done in the actual operational environment or a high fidelity simulation of it.

A.3.8 Configuration Management

1. Configuration management (CM) shall be applied to the objects identified in JPL D-15378, and the approach to CM shall be documented in a CM plan.

CM provides the capability to reconstruct all development artifacts, tools, and products for a previous build or release. Objects requiring CM include the following:

- *Code, including COTS*
- *Build procedures and scripts*
- *Development and test tools -- including operating systems, compilers, assemblers, linkers, design tools, data files that are input to code generators, simulation models, and testbed hardware and software*
- *All test products (plans, procedures, scenarios, cases, data, results, etc) and critical records such as anomaly reports, change requests, and action items*
- *Documentation -- including plans, requirements, designs, release description document, and user guides/helps.*

2. The CM function shall be guided by documentation that describes the scope of CM responsibilities:

- Items to be placed under configuration management
- When each item is to be baselined
- Rules for submitting code to the library, including identification of changes
- Metrics to be routinely collected and reported by CM
- Written procedures needed to implement the CM process.

3. Prior to delivery, CM shall audit the delivery build to verify that this build contains the correct version of each module, that test software has been properly isolated, and that the documentation standards required for delivery have been satisfied.

A.3.9 Software Acquisition

1. When a significant portion of the software for a project is developed by subcontractors or partners (supplier), the Project Implementation Plan shall describe how this software acquisition will be managed.
Pertinent topics to be addressed in a Software Acquisition Management Plan include the following:

- *Provisions for handling requirements change.*
- *Verification by JPL that high-level requirements have been accurately transformed into design requirements.*
- *Verification by JPL of the specification of interfaces between the supplier's product and systems external to it.*
- *Verification by JPL that the supplier's Software Development Plan addresses the software development principles in this document and that the supplier's development plan is being effectively implemented.*
- *Verification by JPL that the supplier's Risk Management Plan is adequate and is being effectively implemented.*
- *Provision for JPL review of any arrangements by the supplier to subcontract or partner a portion of the development for which the supplier is responsible.*
- *Specification of product and process metrics to be reported to JPL by the supplier during development.*
- *Provision for in-process review by JPL of intermediate products -- documents, code, test plans, test results, etc. It is recommended that JPL participate in both milestone reviews and detailed technical reviews.*
- *Verification by JPL that the supplier's Integration and Test Plan is adequate.*
- *Provision for JPL participation in pre-delivery testing and preparation of a JPL plan for acceptance testing of the supplier's completed product.*
- *Identification of documentation to be delivered to JPL by the supplier.*

2. Prior to use in development or integration into the product, a COTS component that incorporates software or firmware (e.g. interface card, micro-controller, FPGA, or gyro) shall be comprehensively tested to ensure that it satisfies documented acceptance criteria and that it contains no anomalies or undocumented features that may constrain its intended use.
In order to identify potential problems in advance, it may be useful to prepare acceptance criteria at the time the purchase order is placed.

A.3.10 Product and Process Verification

1. A project shall establish and maintain a software product and process verification function that covers the full mission software life cycle.
   This function has the following responsibilities:
   - *The documented acceptance of the appropriate products at completion of critical project milestones.*
   - *Audit of the integrity of the software product prior to delivery and at other designated points in the development cycle.*
   - *Periodic verification of the activities and products of the software configuration management function, with special attention to media control, protection against unauthorized access, and back-ups.*
   - *Ensuring that designated intermediate and final software products comply with requirements and standards, and are developed in compliance with documented plans and procedures. This includes in-house development, subcontracted development, COTS software, and COTS hardware components that incorporate software.*
   - *Periodic evaluation of the project's software development process and recommendations for improvement.*

2. All discrepancies found during a software certification review shall be reflected on the Software Review and Certification Record (SRCR) and be recorded as action items that are tracked by the project until closure.
3. Prior to delivery, there shall be an independent verification that all software requirements have been met, that all approved changes have been implemented, and that all anomalies designated for resolution prior to delivery have been resolved.
   Such verification is normally done by an organization that has a reporting channel independent of project management. This verification activity examines test cases and results, and traceability of test cases to software and mission requirements.
4. There shall be a formal acceptance test, involving end-to-end exercise of mission-critical systems, and witnessed by the customer.
   Participation of both the customer and users in acceptance testing is highly recommended.

A.4 Flight Software

1. Prior to computer design/procurement, analysis shall be employed to establish margins for critical performance parameters such as CPU speed, control cycle rates, interrupt rates and durations, communications bandwidth, random access memory (RAM) and erasable programmable read-only memory (PROM and EPROM). Analysis results are documented as the Current Best Estimate (CBE). A development shall observe the following experience-based guidelines for resource margin at critical development milestones:
   - At computer selection, total capability to be: 400% of CBE
   - At implementation start (start of Phase C/D): 60% Margin
   - At launch: 20% Margin*

where Margin = Total Capability - CBE (current best estimate)
% Margin = 100x(Total Capability - CBE)/(Total Capability).

All margin and performance estimates are considered speculative until measured. External instrumentation is recommended.
** To accommodate post-launch fixes, new capabilities, and to maintain adequate in-flight operating margins*

2. The flight software shall be designed to support measurement of computing resources, such as throughput and memory.

3. CBE's for identified margins shall be tracked continually and reviewed at least quarterly as well as at PDR, CDR, ATLO start, and Launch. Margins shall be re-examined in conjunction with proposed significant design changes.

4. Significant deviations from the margin requirements shall be accompanied with both a rationale and recovery/options impacts.

5. Flight software shall accommodate both nominal hardware inputs (within specifications) and transient off-nominal inputs, from which recovery may be required.
Where appropriate, it is wise to include the rationale for a tersely stated requirement so that the designer may understand the broader context in which this component will function.

6. Flight software shall employ appropriate standards in interfacing to the ground or other spacecraft subsystems -- e.g., CCSDS for telemetry and command.

7. All flight software shall be readily modifiable during flight.

8. The attitude and articulation control system (AACS) algorithm and its implementation in flight software shall:
- Be sensitive to identified modeling uncertainties
- Preclude an undesired response to mathematical singularities
- Respond predictably to possible flight events that exceed modeling capabilities.

9. Fault, failure, and anomaly identification and recovery shall be incorporated into the design as early as practical during the design cycle. At the minimum, fault protection software shall be designed to restore -- or maintain if required -- all system functions following or during all credible single-faults.
Additionally, it is good practice to identify non-credible faults at the beginning of design.

10. Redundant processing strings, such as command and data handling, shall be designed to avoid single-point failures that incapacitate all strings -- e.g., an element such as shared memory that is susceptible to corruption or hardware failure.

11. Flight software shall be designed to accommodate processor resets during mission-critical events, such as entry/descent/landing.

12. Firmware incorporated in the command and data handling system shall include error detection and correction (EDAC) logic.

13. Flight computer designs shall include error detection and correction (EDAC) logic on the EEPROMs, and the load process shall be designed to detect and respond to the failure if the EDAC detects an uncorrectable bit error.

14. Flight software shall be designed for testability and operability. Self-test/ diagnostic code shall be designed and incorporated into the software early, so that problem resolution can be done rapidly and the software can be easily adapted by the flight operations team.

15. The software self-test and built-in test routines shall be removable for flight. If not removable, the test routines shall not cause flight hardware damage or interfere with proper execution of the flight software if tests are inadvertently executed.
If built-in test code is removed, prior tests must be rerun to verify that nothing has changed.

16. There shall be at least one dedicated, hardware-in-the-loop testbed for the use of flight software development by a project.
Engineering models of hardware components should be integrated into the testbed as early as practicable.

17. Flight software testbed fidelity shall be maintained. Differences between the testbed and the flight system shall be documented.

18. All critical testing of the spacecraft and mission operations shall be done with the flight version of the software. ("Fly as you test and test as you fly".) If changes are made to the flight software after these tests have been completed, the total test suite shall be repeated.
It is particularly important to test initial condition states, such as boots/resets of both the target system and interfacing systems.
When testing several units of code together, the tester should be aware of the operational use of the modules; run tests as close to real scenarios as possible.

19. Hardware simulation models used in design, unit testing, subsystem integration, and ATLO shall be consistent in order to facilitate meaningful comparisons across platforms. Planned differences in models shall be documented, and differences in expected test results shall be bounded.
In support of software design, models are often developed for sensors, actuators, thrusters, reaction wheels, gimbals, computer buses, etc. Ideally, one should use the same model in all development and test platforms and in ATLO. When this is not possible, it is important to identify model differences and bound the expected differences in test results.

20. When software models or simulations are used to validate a design in lieu of test, these models themselves be validated to ensure that:

- The level of fidelity is appropriate.
- The documented envelope of model validity has been thoroughly verified via systematic parametric variation.
- The uncertainty in simulation results is bounded.
- Sufficient design margin exists to handle the documented model uncertainties.

Document Information / Meta-Data

Sources

Policy: Project Implementation

Document Information

Revision Number	1	**Next Review**	02/16/2002
Effective Date	02/16/2001	**Process**	Develop New Products
Document Type	Guideline	**Process Owner**	Weber, William
DocRev ID	73663	**Document Owner**	Landano, Matthew
Change Description	Section 1.3.6 was added, and Appendix A was modified throughout to incorporate programmatic margins.		

Change History

Revision Number	DocRev ID	Effective Date	Archive Date	Document Owner at Publication	Description
0	64801	02/04/2000	02/16/2001	James Rose	This guideline is an extensive revision to JPL D-17868 in the Vellum File. This is its first version in the DMIE System.