The purpose of this subtopic is to develop technologies, methodologies, and tools that can support the integrated
development of the software system-of-systems necessary for exploration missions. Human space flight challenges
many areas of software technology, including distributed data management and control, sensor interpretation,
planning and scheduling, modeling and simulation, and validation and verification of autonomous systems. This
subtopic focuses on the development portion of the mission life cycle and the dependence of the eventual mission
solutions on the processes and methods used to define and build vehicles and support operations. The need for
such technologies, methodologies, and tools is evidenced by the low success rate of commercial and government
systems, where failure occurs at delivery rather than during operation. Management of the development of such
large systems is essential to integration.

Software Architecture and Systems Integration

The challenges of human system integration for exploration missions is strongly affected by the structure and
architecture of the software systems required to provide control and status pathways to ground support systems
and personnel; to support mission planning and operation; to provide crew interfaces for status, control, and
operation of the vehicle systems, science, and operations, including communications, planning, task management,
interpersonal activity, system configuration management, inventory, food, workflow, resource management,
experiments, and vehicle operation and maintenance. Onboard software must integrate, and be interoperable with
the ground support systems for planning, logistics, operations, science, medical, and engineering, as well as with
subsequent exploration spirals. This requires the development of structures and methods for determining relative
benefits, risks, and costs of the utilization of various engineering approaches. Project management tools are
needed that can conduct and manage Exploration Mission capability and technology gap analysis; provide
technology-to-capability mapping; map technology gaps to research initiatives; and provide decision support.

Systems Engineering Support to Human Systems

There is a need for new tools to support the development of non-avionic control systems throughout the program
life cycle. This includes tools for managing prototyping, requirements, design, design knowledge capture, testing,
and growth and maintenance across multiple development teams. Particular emphasis is placed on design
methods that address the interdependencies between systems. Adapting the Joint Capabilities Integration and
Development System (JCIDS) approach to systems engineering requires development of tools and methodologies
that enable: surveys of current information integration practices between ground-based systems, on-board systems
and crew systems; goal analysis (software task analysis); surveys of existing and proposed technologies; mapping
of technology to tasks; prototyping to drive out design constraints and detailed requirements; development of
testing and evaluation criteria for advanced or untried architectures and technologies and maturation of those technologies into an integrated system of systems; tracking lessons learned, methods, and processes; and development of an experienced personnel base.

Research should be conducted to demonstrate technical feasibility during the Phase 1 contract and show a path toward a Phase 2 demonstration. The contractor will, when possible, deliver a demonstration unit of the hardware and software for NASA testing before the completion of the Phase 2 contract.