The objective of this subtopic is to bring to fruition software engineering technologies that enable engineers to cost-effectively develop and maintain NASA mission-critical software systems. Particular emphasis will be on software engineering technologies for sustaining engineering: achieving affordable reliability over successive spirals of mission software development, maintenance, and upgrades for Crew Exploration Vehicle and Project Constellation. A key requirement is that projects address the usability of software engineering technologies by NASA (including NASA contractors) engineers, and not only specialists.

Many of the capabilities needed for successful human exploration of space will rely on software. In addition to traditional capabilities, such as GNC (guidance, navigation, and control) or C&DH (command and data handling), new capabilities are under development: integrated vehicle health management, autonomous vehicle-centered operations, automated mission operations, and mixed human-robotic teams to accomplish mission objectives. These capabilities will be needed in exploration spirals 2 and 3, including the extended lunar missions. Ensuring that these capabilities are reliable and can be developed and maintained affordably, will be challenging but critical to both the lunar missions and the subsequent Martian missions. Proposals should clearly indicate how the technology is expected to address the challenge of reliability and affordability. Mission phases that can be addressed include not only the software life-cycle (requirement engineering through verification and validation) but also upstream activities (e.g., simulation-based acquisition for software capabilities; mission planning that incorporates trade-space development of software-based capabilities) and post-deployment (e.g., new approaches for computing fault tolerance; rapid reconfiguration, and certification of mission-critical software systems).

Software engineering tools and methods that address reliability for exploration missions are sought. Projects can address technology development and maturation that provide for the following and related capabilities:

- Software-based radiation fault tolerance for computation;
- Methods and tools for development and validation of autonomic software systems (systems that are self protecting and self healing);
• Automated software generation methods from engineering models that are highly reliable;

• Scalable verification technology for complex mission software, e.g., model-checking technology that addresses the 'state explosion' problem and static-analysis technology that addresses mission-critical properties at the system level;

• Automated testing that ensures coverage targeted both at the system level and software level, such as model-based testing where test-case generation and test monitoring are done automatically from system-level models;

• Technology for calibrating software-based simulators and test-beds against high-fidelity hardware-in-the-loop test-beds in order to achieve dependable test coverage; and

• Technology for verifying and validating autonomy capabilities including intelligent execution systems, model-based diagnosis, and adaptive control.

A requirement for a sustainable and affordable human exploration presence in space is the need for modular, reusable elements and subsystems. Major subsystems (e.g., integrated vehicle health management) will present challenges in terms of software-based reconfigurability needed over a long sequence of missions. Projects can address technology development and maturation that provide for the following and related capabilities:

• Software reuse for mission-critical, real-time applications;

• Architectures that facilitate reconfiguration with upgraded components;

• Affordable verification, validation, and certification of upgraded components and sub-systems within a system (or system-of-systems) context;

• Intelligent management of software assets;

• Middleware that enables software platforms to migrate to new hardware platforms (e.g., middleware that enables command and control software to be transparently ported to distributed grid and cluster computer platforms).