xSDK Project Requirements

June 30, 2017

Background

The xSDK Project uses an Agile, Kanban-based software development process. The xSDK Kanban process includes the concepts of Epics and Stories most commonly associated with Scrum. Note that the various xSDK member packages are not required to agree to use this process for non-xSDK work. In line with Agile standards, the xSDK requirements process is lightweight. Below is a description of the strategic objectives and epics for the xSDK project.

Strategic Objectives

The Extreme-scale Scientific Software Development Kit (xSDK) effort has two strategic objectives that drive all project activities, for overarching purpose of supporting the development of high-quality scientific application software.

Build Community

Reusable scientific software packages are developed in many groups. Much of this software can be leveraged by a broader collection of users if the package developers coordinate and collaborate across package teams. The xSDK see community development as a critical means of expanding the usability of individual packages.

Build Sustainability

Reusable scientific software must be sustainable in order for users to rely upon it. Sustainability must be an integral part of the entire software process.

xSDK Epics

Epic 1: Facilitate xSDK Growth

Brief description: Processes and activities related to adding new xSDK packages.

We need a process to identify functionality gaps. Gaps can be satisfied by existing packages to integrate into the xSDK. A gap may require development of a new package. The types of stories that would fall under this epic may include interviewing a stakeholder to determine scientific library needs, or clearly defining xSDK membership requirements for packages.

Epic 2: xSDK Distribution

Brief description: Policies, activities, and tool exploration/development related to distribution, deployment, and installation of the xSDK.

We have two basic types of users. Those who install for themselves and those who install for a group, including an installation at a leadership computing facility. The types of stories that would fall under this epic may include adding a candidate xSDK member package to Spack (the tool used for installing the xSDK), or setting up an installation test on a new LCF machine.

Epic 3: Integration and Use

Brief description: Improve the usability and interoperability of xSDK member packages.

Best practices improve the usability of embedded scientific software, and enable uniformity across independently developed packages. The types of stories that would fall under this epic may include a candidate xSDK member package achieving compatibility with required xSDK Community Policies, improving the interoperability of two or more specific xSDK packages, and adding a page to the xSDK website focused on a topic of importance to developers or users.

Epic 4: High-Quality Product

Brief description: xSDK software should work properly, run efficiently, and solve the required problems.

xSDK must be bug-free on mainstream platforms and must have a comprehensive test suite to isolate errors. The types of stories that would fall under this epic may include setting up an automated test with updated compiler and/or third party library versions, adding a new xSDK Community Policy aimed at compatibility with a recognized software engineering best practice, or documenting an improved process for dealing with regressions found in packages during nightly automated testing.

Epic 5: User Support

Brief description: The xSDK and its member packages should be sufficiently documented and provide clear processes for obtaining support for questions and problems.

User support includes regular updates with bug fixes, tutorials, email contacts and more. The types of stories that would fall under this epic include documenting the process by which xSDK member package teams can add information about their package to the xSDK website, setting up customer email lists, and establishing a service policy containing support expectations and commitments.

Epic 6: Regular Upgrades

Brief description: Improve xSDK capabilities and make the xSDK available on new and existing computing platforms.

The xSDK must have regular releases with the latest features from each package. The xSDK should be available immediately on new platforms so that users can rely on its availability when they port to a new system. The types of stories that would fall under this epic include ensuring the xSDK is usable on a new test bed machine before application teams need access to new platforms, documenting the xSDK release process, and executing the xSDK release process.