4.2.7 Manageability

Return to Non-Functional Requirements

- ☐ **[char]Please Review**

Manageability is most important during the second half of a System Lifecycle phases (i.e. operation, maintenance, support). Manageability can greatly influence the recurring costs and can increase the chances of a failure. Often a system that hard to manage is described as fragile since the smallest of changes can have dire consequences on the systems functionality.

> *Manageability directly influences a system's reliability, availability, security, and safety, thus being a key ingredient of system dependability.*
>
> *Just like security and safety, manageability is generally hard to retrofit in complex systems—it is always easier to build it in from day one. However, in the absence of means to measure manageability and to quantify the various tradeoffs, it is difficult to get the design right. We proposed a manageability metric that combines management workloads and weightings based on real world studies with direct measurement of the number of steps involved in management tasks and their duration.* [1)]

- 02_types
- 04_costs
- 06_system
- 08_software

Software Manageability Issues

Return to the Top

Over the last few decades, many advances have been made in terms of Open Source Software (OSS) which has to change the way that software was developed, released and used. During this same time period, the traditional Waterfall Model of System and software development has also been largely supplanted with the Agile Model. Many of these changes also have to do with the evolution of systems (or projects) from being Greenfield to Brownfield development and from a “build the world” attitude towards “integrate and glue the world” mindset.

Successful OSS development and adoption not only has to produce products which are solid, strong and robust but also must meet the needs of a Community of Interest (CoI) that has coalesced around a single minded, purpose built, functionality (i.e., Apache Tomcat application server, PostgreSQL Database, Node.js an asynchronous event-driven JavaScript runtime, Docker containerized apps, Kubernetes orchestration engine for containers, etc.). Many of the OSS products are part of many of the successful projects today.

However, its not good enough to just write software and make it publicly available. At the heart of these successful efforts are the well governed, focused, supporting CoIs. There is a desire from almost all systems (or projects) to join the OSS trend but unfortunately, the need for strong governance and rigorous methodology is minimized or skipped in the name of expediency. Fortunately, there is an organization which can help with this called Talk Openly Develop Openly (TODO) (not to be confused with a `to-do`).

TODO organization, though focused on OSS, has written a series of white papers that are well worth studying and using even it your system (or project) is not OSS. One of these papers which is particularly germane to Software Manageability is Tools for managing open source programs [2)]. It is beyond the scope of this document to try to recreate the full content of this white paper. It does present a list of many of the tools available for managing software and how to use them. Here is the table of content from the document:

- Why you need special tools for open source program management
- How to select and plan your tools
- Elements of a basic toolset
- Tools for managing source code
- Tools for tracking project health
- Tools for communications and collaboration
- Tools for corporate-scale GitHub management

DDS Specifics

Return to the Top

Data Distribution Service (DDS) can not solve all of a systems(or projects) Maintainability issues, however, by DDS's design, it can eliminate or reduce the Manageability issues that could arrise from using DDS.

Table 1: DDS role in helping Manageability

Kinds of Management	Description
Health Monitoring, Logging, and Alerting	Although there are currently no DDS standards for that directly supports System Monitoring, each of the DDS Vendors have sets of tools which can be used for that purpose. These tools include • Development and troubleshooting including specialized network sniffers, modeling tools, and code generators • System monitoring and administration including terminals, shared memory management tools, recorders and replayers • Functional, systems and performance testing • Federated Discovery • Bridges to other Message-Oriented Middleware (MOM) products • Topic aggregators
Configuration and Control	DDS uses a standardized Discovery process which eliminates most of need for configuration. Some DDS Vendors offer specialized or advanced tools that aid in tuning DDS configurations and discovering performance issues
Deployment and Updates	DDS Extensible and Dynamic Topic Types for DDS (DDS-XTypes) allows for planned evolution of the Datatypes within a ddsapplication. For example, adding or removing fields in a Data Structure, changing the basic type from an int16 to an int32, etc.
Asset Discovery and Inventory	DDS automatically registers all Data Writer and Data Reader allowing them to be discovered. It is possible

1)

Toward Quantifying System Manageability, George Cadea, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, Accessed 20 July 2020, https://www.usenix.org/legacy/event/hotdep08/tech/full_papers/candea/candea_html/index.html

2)

Tools for managing open source programs, Talk Openly Develop Openly (TODO), Accessed 20 July 2020, https://todogroup.org/guides/management-tools/