The Buzzword Puzzle: Finding the Right Technology for the Problem

Todd A. King[1], Raymond J. Walker[1], and Steven P. Joy[1]

[1]Institute of Geophysics and Planetary Physics, UCLA, 3846 Slichter Hall, Los Angeles, CA 90095-1567 USA

1 Understanding The Problem

2 System Engineering

System engineering is the process of managing the system's life cycle through concept, design, production, operation and disposal. The primary purpose of systems engineering is defining and characterizing the behavior and interaction between system components. The more complex the system is the more challenging is its engineering.

The system engineering process involves the following:

- requirements capture
- systems architecture and design
- functional analysis
- interface design and specification
- communications protocol design and specification
- simulation and modeling
- acceptance testing
- verification, validation and fault modeling

3 System Engineering Methodologies

Whether a project is large or small some form of system engineering takes place. For small, highly focused projects the system engineering can be informal with little or no written documentation. For complex projects which consist of multiple, interacting components a formal methodology is followed to enable proper management of the project. These methodologies fall into two broad categories:

Adaptive methods focus on adapting quickly to changing realities. Some examples are:

Agile Programming: Developing software in short timeboxes, called iterations, which typically last one to four weeks. Some examples are:

Extreme Programming (XP): Start simple and refactor as necessary.

Adaptive Software Development: A repeating series of speculate, collaborate, and learn cycles

Predictive methods focus on planning the future in detail. Some examples are:

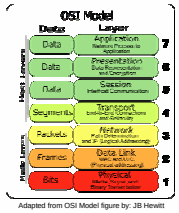
Waterfall: Development moves steadily through phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance.

Spiral Model: Development occurs in phases that start with a design goal and end with the client reviewing the progress. Analysis and engineering efforts are applied at each phase of the project.

Cowboy coding is the absence of a defined method: team members do whatever they feel is right.

4 The Importance of Models

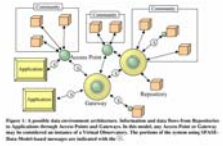
A model presents a vision for an eventual implementation. It conceptualizes distinct aspects of system or process. Models are useful in comparing different implementations. In practice an implementation may not be divided into discrete components that correspond with each unit of a model. For example, many internet applications span multiple layers in the OSI model.



Adapted from OSI Model figure by: JB Hewitt

OASI Functional Model
(Open Archival Information System)

SPASE System Model
(Space Physics Archive Search and Extract)



5 Requirements

Requirements analysis is a term used to describe all the tasks that go into the instigation, scoping and definition of a new or altered computer system.

The benefit of requirements analysis is to reduce both risk and cost. The difficulty of requirements analysis is finding people with adequate experience, technical expertise, and language skills to perform the task. Stakeholders (end-users) may have difficulty quantifying their needs and expectations and software developers may lack the necessary domain knowledge and have difficulty understanding the business process.

Requirements analysis can be a long and arduous process. Requirements workshops, prototyping and use cases can aid in identifying requirements.

Adaptive system engineering methodologies have less formal requirements gathering and are most appropriate for projects with:

- Low criticality
- Senior developers
- High requirements change
- Small number of developers

Predictive system engineering methodologies have more formal requirements gathering and are most appropriate for projects with:

- High criticality
- Junior developers
- Low requirements change
- Large number of developers

Abstract

What is the best way to move data from point A to point B? That all depends on what connects the two points and what kind of data you are trying to move. Selecting the right technology requires a careful consideration of the problem, an awareness of existing solutions and knowledge of appropriate standards. When moving a solution through the design, implementation and deployment stages there are a variety of development methods. Matching the development method to the scale of the task is important. Some of the technology challenges that the science community must address overlap with our commercial counterparts. However, there are some technology challenges which are unique to science and require new and innovative approaches. Based on our experience developing solutions for NASA's Planetary Data System (PDS), Dawn Science Center, and the Space Physics Archive Search and Extract (SPASE) project we have considerable experience in what does and does not work. We present an assessment of the adoption threshold of selected standards, technology and development methods. We will discuss how changing expectations and evolving requirements impact the effectiveness of chosen solutions.

6 Implementation

7 Overview

Many technical challenges that are faced during software implementation have been encountered in a different context and a solution has been developed. If each project approaches their implementation as completely unique then its resources will be spent to re-invent. Innovation is incremental improvement. Paradigm shifts are rare.

8 Guiding Principals

A set of principles to guide the selection:

1. Adopt a standard if one exists.
2. Use an existing protocol if it solves the problem
3. Refactor when ever possible.

9 Adoption Threshold

The adoption threshold is where the cost of adoption is at or below a desired price point. This economic principle can be applied to software development by asking what level of effort is required to deploy a solution. If your application will run on selected platforms the adoption threshold (and development costs) will be lower if intrinsic services are used. The determination of the adoption threshold is partially subjective and partially objective.

Some examples are:

File Transfer

method	threshold	Comments
http	low	Simple, widely supported.
Proprietary	high	Requires specialized support.

Information Encoding

method	threshold	Comments
XML	medium	Schema must be developed
Proprietary	high	Requires specialized support.

10 Changing Expectations

When a system takes shape users and developers often envision new capabilities. Adopt a system engineering method that is aligned with the anticipated rate of expectation evolution. If rapid evolution is expected use an adaptive methodology. If evolution is anticipated to be slow or non-existent use a predictive methodology.

11 Assessing a Solution

Evaluation is an important task when an implementation is complete. The basic evaluation dimensions are:

effectiveness: Can users use it? Are costs reasonable?

reliability: What is the confidence that the system works as it should?

resiliency: What happens when things go wrong.

12 The Importance of Standards

Standards are published specifications (use, rules, guidelines or characteristics) that provide a common platform for both individual and general benefit. A standard enables independent development of cooperative systems and the sharing of information. Standards are used to describe expectations and to determine if expectations are met.

13 Who Defines Standards

ANSI: American National Standards Institute. Founded 1918. The U.S. member of ISO.

IETF: Internet Engineering Task Force. Founded 1986. The IETF is the descendant of the Request for Comment (RFC) method of documenting "standards" originating with the ARPANET in 1969. http://www.ietf.org/rfc.html

IEEE: Institute of Electrical and Electronics Engineers. Founded January 1, 1963.

ISO: International Standards Organization . Founded February 23. 1947. http://www.iso.org/

W3C: World Wide Web Consortium. Founded in 1994. http://www.w3c.org/

Your neighborhood working group like PDS or SPASE.

PDS: http://pds.nasa.gov/
SPASE: http://www.igpp.ucla.edu/spase

14 A Protocol For Every Need

When developing cooperative or distributed applications information is exchanged by using a protocol. A protocol is a set of standard rules for data representation, signaling, authentication, and error detection required to exchange information. There's a protocol for almost every need. Some protocols can easily be refactored for new applications. Some protocol examples are:

File Transfer

FTP File Transfer Protocol
HTTP HyperText Transport Protocol, used in the Web
POP3 Post Office Protocol Version 3
SMTP Simple Mail Transfer Protocol
IMAP Internet Message Access Protocol
WebDAV Web Distributed Authoring and Versioning
NNTP News Network Transfer Protocol

Services

SNMP Simple Network Management Protocol
SSH Secure SHell
NTP Network Time Protocol
LDAP Lightweight Directory Access Protocol
IRC Internet Relay Chat Protocol
DICT Dictionary protocol
RADIUS An authentication, authorization and accounting protocol
SOAP formerly Simple Object Access Protocol, now just SOAP.

15 Summary

- A thoughtful consideration of the goals and expectations for a project is critical in selecting the best technology for the task.
- Existing technologies which can be adopted or refactored should be selected over any first principal development.
- Awareness of existing standards and protocols contributes to reducing both the cost and risk of software development.