A good monitoring system should have or been:

**Universal Monitoring**

We don’t want to implement a monitoring solution for the Linux server, another for network equipments, another for hardware, etc. We need one monitoring solution to monitor almost all our IT systems and services. A good monitoring system should provide a framework to include plugins to monitoring various services and devices. For example, it should be able to monitor: Operating systems *nix, Windows, etc. System resources CPU, Disk, Swap, Process, etc. Network Equipments Switches, Router, VPN, Firewalls, UTL’s, PLC’s etc.

**Efficient Alert Notifications**

The system should be able to assign individuals (or groups) to a system or service as owners. This gives the power to the owners. Let the owners of the system (or service) be notified and take action. Should provide the ability for us to send notification using various methods email, pager, SMS, IM, etc. Ability to set warning and critical alerts for systems and services that are monitored. Granular monitoring options to specify how often the system should be monitored, how many retries in case of failure, how many failure notifications to send, methods of notification, etc.

**Web Dashboard**

that provides overall health, issues, and alerts for all the systems across the network, along with the ability to drill-down to individual hosts (and services).

**Issue Escalation**

Should provide the ability to notify managers, when the owner of the system is not taking action on an issue within certain time period. For example, when a PLC crashes, and the responsible doesn’t fix it within reasonable time, the monitoring system should alert the manager about the issue.

**Distributed Monitoring and Scalability**

Should be capable of monitoring thousands of servers and services without too much overhead. Support distributed monitoring with multiple monitoring systems across the enterprise that can talk to a central monitoring server.

**Reporting**

Should generate various monitoring reports. For example, availability, trending, notification reports for administrators. Should provide daily, weekly, monthly, or custom date range analysis of various monitoring statistics.

**External Application Integration**

Should provide a framework (or API) that can be used by external application to update the current status of the system or service that is getting monitored. Should be able to provide enough details for external vendors to integrate their solution with the monitoring software. The more extensible the framework is, more vendors will provide solution, and more companies will use it to make the software robust.
Open source solution

Since we'll be exposing all our mission critical systems to the monitoring software, we should make sure that you can trust the monitoring software. Open source solutions are typically thoroughly tested and reviewed by the community for any potential security issues. Look for the track-record of the software. How many years it has been in the market, the longer the better. How many companies are using the software, the more the better.

Community and Commercial Support

When we are implementing it on a large scale (thousands of servers), we might want to implement a solution that is official supported and backed by a company. Several open source monitoring solutions are backed by a company that provides commercial support. Even if you don't use the commercial support initially, you might want to use the support, when you expand your monitoring footprint.

Easy to Learn and Use

This might be obvious to some of us, but we'll be surprised how many people end-up implementing a system that is very hard to learn and use. Don't overlook this. The monitoring solution should be easy to implement and learn, as simple as that. We should not spend weeks trying to figure out how to get the software implemented and working successfully.