1 Introduction

Quality has many factors/attributes:

1. accuracy: results of process are accurate.
2. completeness: results of process are complete.
3. efficiency: the process is efficient, time, space, etc.
4. usability: usable by users
5. satisfaction: users are satisfied, trust the results.
6. flexibility: flexible to future needs
7. compatibility: interoperability with other systems
8. portability: can be ported to other environments.
9. maintainability: easy to maintain
10. mitigate risk: unlikely to unexpectedly blow up; will your software kill anyone?
11. reliability: fault tolerant
12. security: secure from unauthorized uses; keeps data private, etc.
13. availability: less downtime
14. scalability: double the data, double the runtime?

Assessment is a process to see where software stands in relation to above factors.

2 Cost of Quality

Due to cost of quality, software that is often "good enough" gets released into production. There needs to be a balance: Cannot ship unfinished stuff, and cannot polish things forever due to missed opportunities.

3 Monitoring

Measure everything measurable. Explain outliers. If something cannot be measured, then measure a proxy (e.g. cannot measure customer satisfaction, so measure the amount of time customers spend on the website, etc.). If you notice an outlier, investigate: might be a symptom of an issue. If get too many false positives, adjust thresholds.

Measure things in production. Run sanity checks in production, to ensure unexpected things get noticed quicker, and corrected. Often overlooked "thing" that may happen is that the data was not uploaded or updated (not that the data is wrong: it may be fully missing).

For online systems, prioritize errors that end user interaction/transaction sessions. For example, ordering an airline ticket, and website bombs out with "unspecified error", repeat next day, it bombs out with "unspecified error" again; these sorts of things should be monitored for and corrected—since the company is losing a sale.