

Quality Management II

<http://lbgeeks.com/gitc/pmQual.php>

June 10, 2008

- **Quality Documentation II**
- **Special Concerns For Software**
- **Summary**

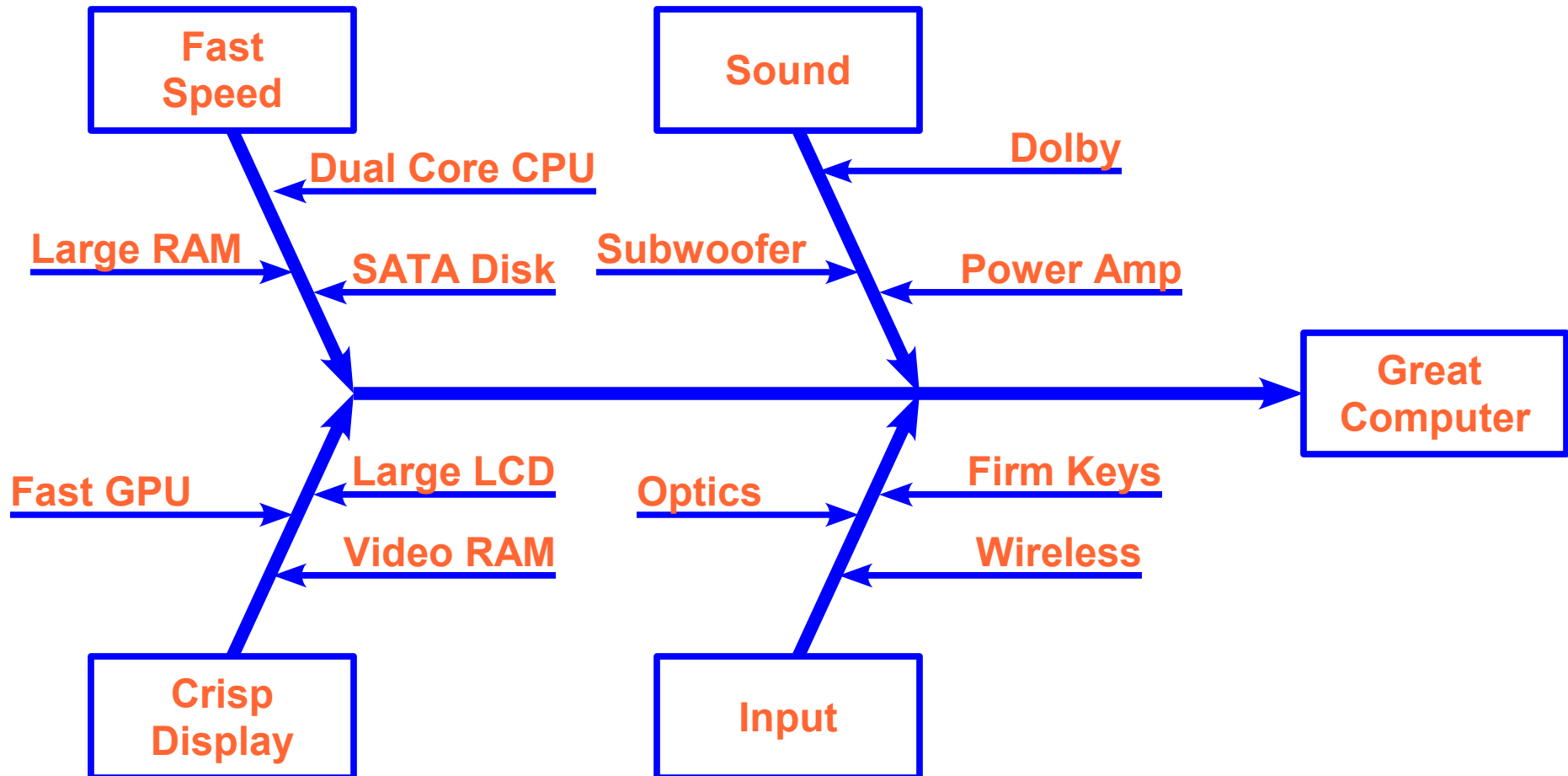
Quality Documentation

- **Phase I:**
 - **Check Sheet**
 - **Histograms**
 - **Control Charts**
- **Phase II:**
 - **Ishikawa Diagrams**
 - **Pareto Chart**
 - **Scatter Diagrams**
 - **Flow Chart**

Ishikawa Diagrams

- **Cause and effect or “fish bone” diagram**
- **Positive and negative directions:**
 - **What makes a great sports car?**
 - **Why do buyers say a car has poor quality?**
- **Factors organized into groups**
- **Each group has a hierarchy**
- **Enable team collaboration**

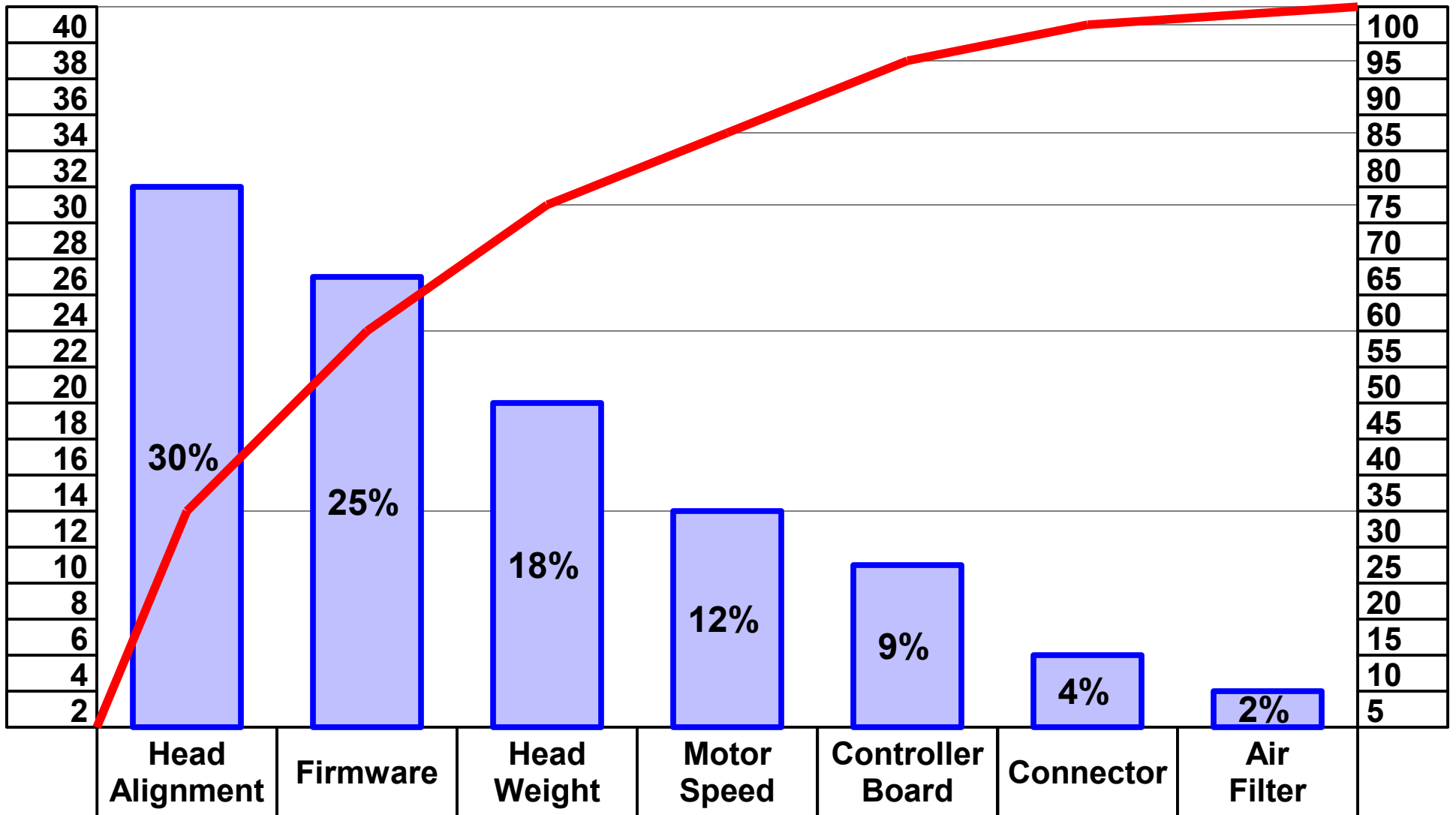
Ishikawa Example



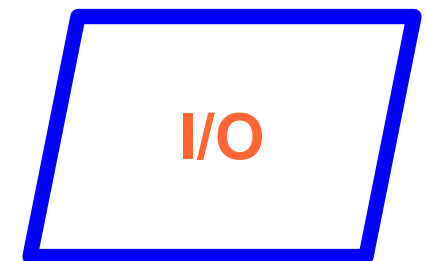
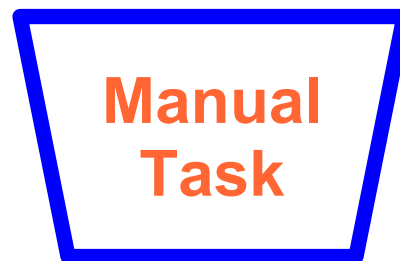
Pareto Charts

- **Another cause and effect tool**
- **Corresponds failure with probability**
- **Bar chart:**
 - **X axis failure reason**
 - **Y axis percent occurrence**
- **Line graph of cumulative percentage**
- **Helps set priority for spending on improvement efforts**

Pareto Example



Flow Charts



Special Concerns For Software

- **Requirements Phase**
 - **Separating Requirements and Design**
 - **Derived Requirements**
 - **Requirements to Design Traceability**
- **Developer Testing**
 - **Prototyping**
 - **Unit Testing**

Separating Requirements and Design

- **Design must follow from requirements**
- **Reversing this order has consequences:**
 - **Failure to meet requirements**
 - **Increased development cost**
 - **Inefficient operational cost**
- **Design imposed during requirements phase is called a constraint**
- **Constraint value (benefit – cost) must exceed those of alternatives**
- **Constraints can limit product quality**

Derived Requirements

- **Direct requirement is stated by someone**
- **Derived requirements follow from:**
 - **Necessity of direct requirement**
 - **Social and legal issues**
 - **Other constraints, such as reuse demand**
- **Impact on quality is significant:**
 - **Law may require a QA department**
 - **Existing method may impact precision**
- **Specific coding or testing methods may be derived from quality specification**

Requirements to Design Traceability

- **Each design element should correspond to at least one requirement**
- **Lapse of quality if requirement has no impact on design**
- **Traceability matrix links requirements to design elements:**
 - **Requirement has unique ID (primary key)**
 - **Design element cross-references ID**
 - **Forward and backward integrity checks**

- **Customers care about non-functional requirements:**
 - **Performance, use of CPU, RAM, disk**
 - **Reliability, probability software will fail**
 - **Maintainability, detected errors easily fixed**
- **Often multiple design choices:**
 - **Arrays**
 - **Lists**
 - **Hashing**
- **Developing a small prototype helps evaluate alternatives, increase quality**

Unit Testing

- **Object should control its own state:**
 - No independent change of member data
 - Incorrect parameters should cause no harm
 - Methods must be atomic
- **Difficult to predict how an object might be actually used:**
 - Multiple possible operation sequences
 - Lower level objects used to build systems
- **Unit test should exercise:**
 - All private and public members
 - Boundary conditions for valid, invalid data
- **Regression test if object ever updated**

Summary

- **Quality management is necessary to keep projects on track**
- **Most techniques stem from production: repeated processes**
- **Collection and retention of data from project processes necessary**
- **Documentation is specialized for quality management purposes**
- **Software requirements, testing phases have unique impact on quality**