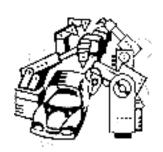
Manufacturing Quality

18-849b Dependable Embedded Systems
Michael Scheinholtz
January 28th, 1999



Carnegie Mellon

Overview: Manufacturing Quality

Introduction

Key concepts

- The Manufacturing Paradigm
- Manufacturing from an embedded systems perspective
- Manufacturing software
- Quality

Relationship to other topics

- Robustness
- Reliability
- Business model

Area of Exploration

Can high quality manufacturing lead to high quality software?

Why should we care about Manufacturing?







- Affects cost, reliability, and robustness
- **◆** Maybe we can learn from its methods

- Interchangeable parts
- Mass Production
- **◆ Total Quality Management**
- **◆ Six-sigma Quality**

Interchangeable parts

- Traditional approach
 - each machine is made from scratch, with a unique set of parts
- New idea, circa 1900
 - make machines from common parts
 - » eased construction and replacement
 - » a machine could be made using one construction method
 - » led way for Henry Ford and Mass production

Mass Production

- Henry Ford created a factory to mass produce Model-T's in 1913
- Bring the foundation to the parts
- Labor is interchangeable and specialize
- This method works for food too (McDonalds), how about software?

Quality Engineering



DILBERT Copyright 3 1998 United Feature Syndicate, Inc. Redistribution in whole or in part prohibited.

◆ STDM, VOC, CAD/CIM, ISO 9001, and UATAQ

- (Using Acronyms To Achieve Quality)
- TQM (Total Quality Management)
 - Quality issues exist in every part of a corporation, so apply quality everywhere
- **♦** Measuring Quality: defined from the end user's perspective
- Design to Manufacture
 - From CAD directly to the manufacturing floor.

◆ Six-Sigma Quality (Motorola Inside[™])

- The mean value of an operating parameter may vary as much as 1.5σ .
- 6σ quality means a product will work when its operating parameters are within 6σ of the norm.

- Achieving Six-Sigma requires both quality in design both and manufacturing
- Process driven method

Embedded Systems Considerations

Cost

- Design For Assembly
 - Minimize resources needed to assemble (motion, locality)
- Design in high tolerance for variation

Bringing the factory to the software industry

Software Factories in Japan

Cleanroom Software Engineering

- Can factory methods be moved to the software house
 - Design to manufacture? (Automatic program generation)
 - Reusable/interchangeable parts?
 - Total Quality assurance

Software Factories

• Japanese combining western software engineering methods with their own manufacturing quality techniques.

Features

- All phases of software creation done in a unified environment
- Reusable components (trying to automate this)
- Assembly line organization (design separated from test)
- Quality checks: specified number of faults allowed per each phase
- Group peer reviews

Research Areas

- Design to Manufacture: code generation from a specification
- computer assisted programming

Results

claim from Toshiba Fuchu Software Factory: for real-time control system
 0.3-0.1 faults per KLOC

Cleanroom Software Engineering

- Separation of design and test (factory line organization)
 - Designers are not allowed to compile code, but all code is subject to intense peer review.
- Formal specification as much as project allows
- Statistical testing to certify quality done from user's perspective
- Incremental additions always done at the user level
- Different groups adapt different pieces as needed

Results

- IBM: COBOL/SF added 52,000 lines of code to 40,000 base, 179 errors (norm 1500 - 3000), 740 LOC/LM (norm 150)
- NASA: training was 4% of project hours, 69% higher productivity, 45% reduction in error rate

Conclusions & Future Work

- What should an Embedded systems engineer worry about?
 - Manufacturing quality control
 - Manufacturing costs
- What can we take from the factory?
 - Separation of design and test
 - Get it right in the design phase
 - Design to manufacture won't happen for a long time

Paper: Cleanroom Software Engineering

How useful is the Cleanroom method?

Are their any other areas of manufacturing that can be applied to software engineering?

