“If builders built buildings the way programmers write programs, then the first woodpecker that came along would destroy civilization.”

– Gerald M. Weinberg/Weinberg’s Second Law
Software Quality Assurance (SQA/PPQA)

- **Anti-Patterns:**
  - Process steps are being skipped
  - Nobody tasked with ensuring SW process is actually being followed
  - Less than 5%-6% of effort on SQA

- **SQA / PPQA* responsibilities:**
  - Define & maintain software process definition
  - Train on process and related skills
  - Audit to ensure process is being followed
  - Keep metrics & diagnose process-related failures
  - Coach/intervene to fix process failures

* PPQA = Product and Process Quality Assurance
Every process step produces artifacts
- If it’s not written down, it didn’t happen
  - Audit says “yes, it happened”
- Artifacts must be lightweight but useful
  - Artifacts are the arrows on V diagram

Process quality audits are more subtle
- Some information from quality of written artifacts
  - Completeness – were all required fields filled out?
  - Internal consistency checks, e.g., traceability
- Some information indirectly via process-related metrics
  - Are defects escaping to later process stages?
- Sometimes direct observation (e.g., randomly attend peer reviews)
SQA Personnel As Coaches

- **SQA defines the process**
  - With inputs from all stakeholders!
  - Keeper of the process diagram
  - Create/maintain templates and work aids

- **SQA conducts training**
  - Initial training for new team members
  - Remedial coaching, guidance, etc. for process failures

- **SQA keeps records & metrics to ensure process is on track**
  - Metrics should encourage high quality work products
  - Better to coach problems than punish them

One Reason Process Matters

- **Poor process:**
  - Bugs escape to field

- **Testing:**
  - Can’t even look for bugs until it’s **expensive** to find and fix them

- **Peer Reviews:**
  - Find bugs while it is still **cheap** to do so

---

**Relative Cost to Fix Defects**

- **Source:** Davis 1993

- **Product Development Stages**
  - Requirements
  - Design
  - Coding
  - Unit Test
  - System Test
  - Maintenance

- **Cost Multipliers:**
  - Requirements: 1x
  - Design: 5x
  - Coding: 10x
  - Unit Test: 20x
  - System Test: 50x
  - Maintenance: 200x
TOYOTA’S INADEQUATE SOFTWARE PROCESS

- Toyota failed to exercise a safe standard of care for software
  - Relied too much on vendors
  - Lacked internal expertise
  - Inadequate supervision and training of software

There is a process in place for hardware, but not software:
- Process IN/OUT, timeframe not decided.
- There are processes at Toyota yielding no knowledge.
  ⇒ Large deviation in product quality among vendors.

Barr Chapter Regarding Toyota’s Code Complexity
SQA Best Practices

- SQA must have broad view of quality
  - Define & maintain processes
  - Audit & measure process effectiveness
  - Train & intervene to keep process conformance on track
  - About 5%-6% of staffing for SQA

- SQA pitfalls
  - SQA is not testing (product defects); it’s about process defects
  - Avoid form over substance; technical quality matters!
  - Beware adversarial role of SQA (the “Process Police”)