Fourth Lecture Hour
8:30 – 9:20 am, September 9

The Old Way and the New Way

(Chapter 4 of Royce' book)

Topics for Today

- The Principles of Conventional Software Management
- The Principles of Modern Software Management
- Transitioning to an Iterative Process

Review - Quality Improvements with a Modern Process

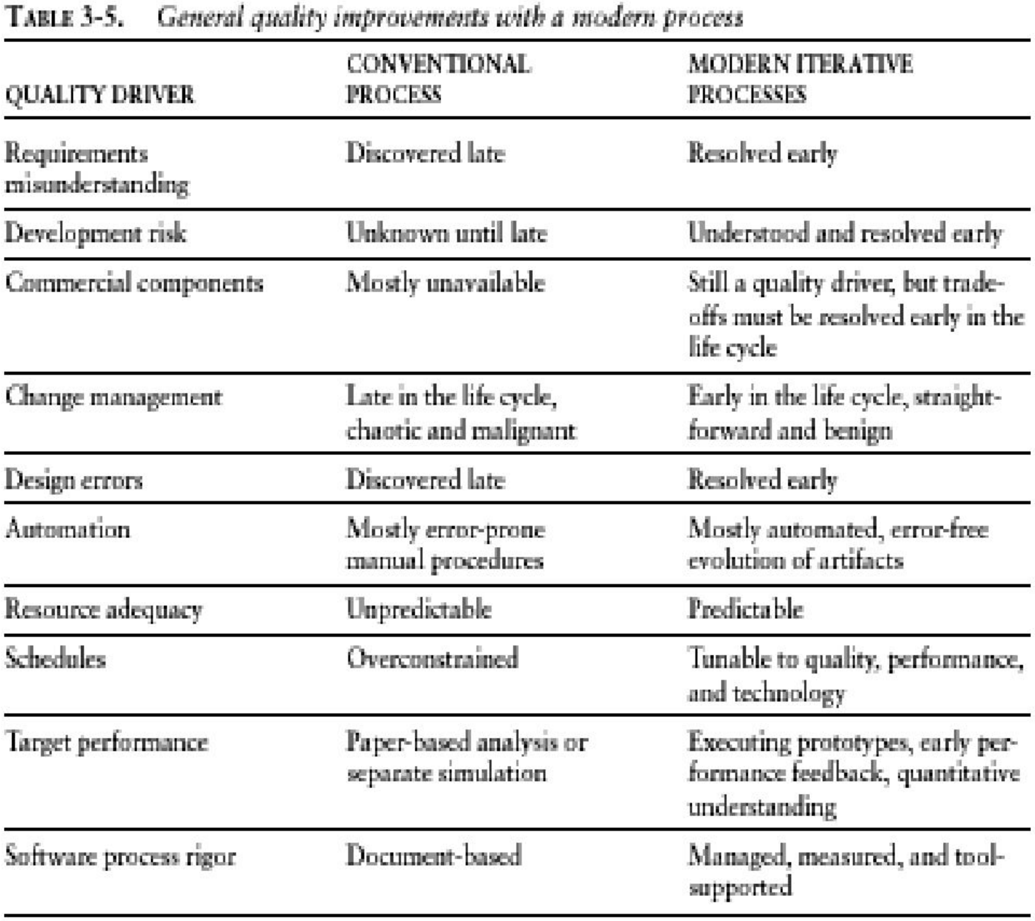


TABLE 3-5. *General quality improvements with a modern process*

QUALITY DRIVER	CONVENTIONAL PROCESS	MODERN ITERATIVE PROCESSES
Requirements misunderstanding	Discovered late	Resolved early
Development risk	Unknown until late	Understood and resolved early
Commercial components	Mostly unavailable	Still a quality driver, but tradeoffs must be resolved early in the life cycle
Change management	Late in the life cycle, chaotic and malignant	Early in the life cycle, straightforward and benign
Design errors	Discovered late	Resolved early
Automation	Mostly error-prone manual procedures	Mostly automated, error-free evolution of artifacts
Resource adequacy	Unpredictable	Predictable
Schedules	Overconstrained	Tunable to quality, performance, and technology
Target performance	Paper-based analysis or separate simulation	Executing prototypes, early performance feedback, quantitative understanding
Software process rigor	Document-based	Managed, measured, and tool-supported

Thirty Principles for Conventional Process (Davis)

- 1. Make quality #1
  - Understand early the tradeoffs among features, quality, cost and schedule.
- 2. High quality software IS possible.
  - Prototype, simplify, involve the customer.
- 3. Give products to the customer EARLY.
  - Determines the REAL requirements.
  - Use prototypes, demonstrators, alpha/beta releases.

Thirty Principles for Conventional Process – cont’d

- 4. Determine the problem BEFORE writing the requirements.
  - Resist jumping to solution.
- 5. Evaluate design alternatives.
  - Decouple architecture from requirements.
- 6. Use an APPRORIATE process model.
  - Considerations include corporate culture, risk tolerance, and volatility of requirements.

Thirty Principles for Conventional Process – cont'd

- 7. Use different languages for different phases.
- 8. Minimize intellectual distance.
  - Use real-world structures.
- 9. Put techniques before tools.
- 10. Get it right before you make it faster.
  - It is far easier to make a working program run faster than it is to make a fast program work.

Thirty Principles for Conventional Process – cont'd

- 11. Inspect the detailed design and code early.
- 12. Good management is more important than good technology.
  - The best technology will not compensate for poor management.
- 13. People are the key to success.
  - The right people, even with insufficient tools, languages and processes will succeed.
- 14. Follow with care.

Thirty Principles for Conventional Process – cont’d

- 15. Take responsibility.
  - It takes more than good tools, methods and components. It also takes good people and good management.
- 16. Understand the customer’s priorities.
  - Although the customer may not always be right, the customer is always ready to understand.
- 17. The more the customer sees, the more the customer needs.
  - Software manager needs to have objective data to argue change requests – to balance affordability, features, and risk.

Thirty Principles for Conventional Process – cont'd

- 18. "Plan to throw one away."
  - Rather, plan to evolve from prototypes to final product.
- 19. Design for change.
  - Architecture must accommodate change. Think ahead!
- 20. Design without documentation is not design!
  - However, engineering work should be largely self-documenting.

Thirty Principles for Conventional Process – cont'd

- 21. Use tools, but be realistic.
  - Modern, iterative development methods require extensive automation.
- 22. Avoid tricks.
  - However, be mindful and responsive to innovation.
- 23. Encapsulate.
  - Think in terms of component-based and object-oriented design.
- 24. Use coupling and cohesion.
  - Cohesive components with minimal coupling are easier to maintain and adapt to changes.

Thirty Principles for Conventional Process – cont’d

- 25. Use complexity measures.
  - Royce recommends McCabe.
- 26. Don’t rely on testing your own software.
- 27. Analyze causes for failures.
- 28. Realize that software entropy increases.
  - Software grows more complex and more disorganized over time.

Thirty Principles for Conventional Process – cont'd

- 29. People and time and NOT interchangeable.
  - Read the Mythical Man-Month.
- 30. Expect excellence.

Modern Software Management

Royce Top Ten Principles

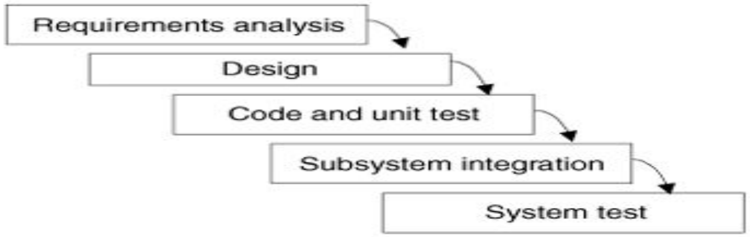
The First Top Five Principles for a Modern Process

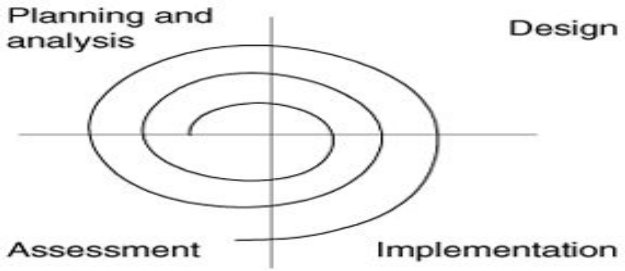
Waterfall Process

Requirements first
Custom development
Change avoidance
Ad hoc tools

Iterative Process

Architecture first
Component-based development
Change management
Round-trip engineering





Architecture-first approach → **The central design element**

Design and integration first, then production and test

Iterative life-cycle process → **The risk management element**

Risk control through ever-increasing function, performance, quality

Component-based development → **The technology element**

Object-oriented methods, rigorous notations, visual modeling

Change management environment → **The control element**

Metrics, trends, process instrumentation

Round-trip engineering → **The automation element**

Complementary tools, integrated environments

First Five Improvement Principles

- Architecture first approach.
  - Balance driving requirements, architecture and design decisions, and life-cycle plans.
- Iterative life-cycle process to confront risk early.
  - Refine problem and solutions over several iterations.
- Use component-based development.
  - Move from “line-of-code” mentality to “component” mentality.
- Establish a change management environment.
  - Important for iterative development.
- Use round-trip engineering.
  - Automation of change management, documentation and testing across requirements, specifications, design models, source code, executable code, and test cases.

Remaining Five Improvement Principles

- Capture design artifacts with model-based notation.
  - Far more objective process than using human review and inspection processes.
- Use objective quality control and progress assessment.
  - Assessment should be integrated with the development process, not ah-hoc when difficulties occur.
- Use a demonstration-based approach to assess intermediate artifacts.
  - Apply to early prototypes, baseline architectures, early releases.
- Plan on intermediate releases with evolving levels of detail.
  - Make early and continuous releases with realistic use cases and scenarios.
- Establish a good configurable process.
  - Must be economy of scale and be scalable across a range of projects.

Modern Approaches for Solving Conventional Problems

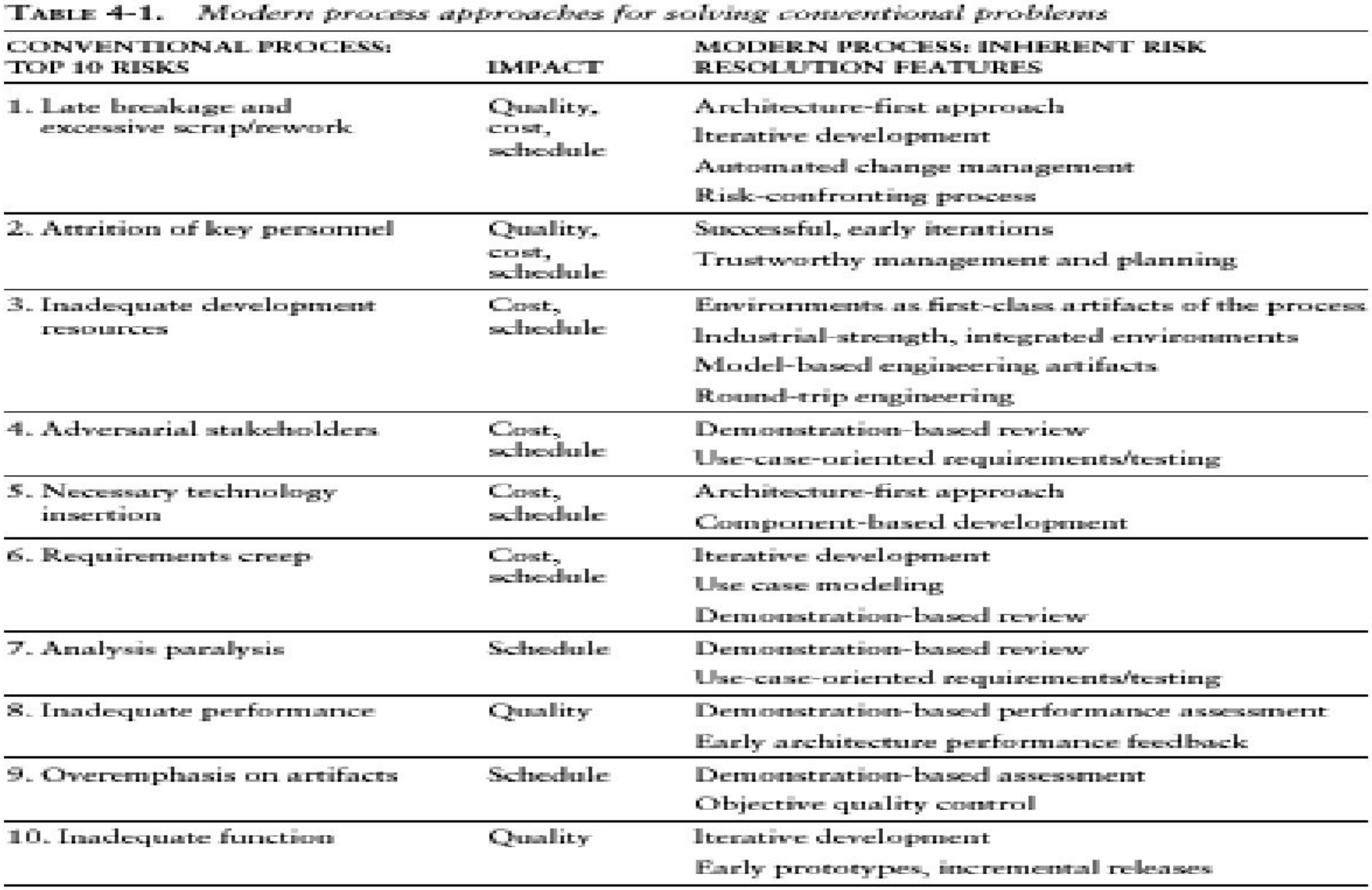


Table 4-1. *Modern process approaches for solving conventional problems*

CONVENTIONAL PROCESS: TOP 10 RISKS	IMPACT	MODERN PROCESS: INHERENT RISK RESOLUTION FEATURES
1. Late breakage and excessive scrap/rework	Quality, cost, schedule	Architecture-first approach Iterative development Automated change management Risk-confronting process
2. Attrition of key personnel	Quality, cost, schedule	Successful, early iterations Trustworthy management and planning
3. Inadequate development resources	Cost, schedule	Environments as first-class artifacts of the process Industrial-strength, integrated environments Model-based engineering artifacts Round-trip engineering
4. Adversarial stakeholders	Cost, schedule	Demonstration-based review Use-case-oriented requirements/testing
5. Necessary technology insertion	Cost, schedule	Architecture-first approach Component-based development
6. Requirements creep	Cost, schedule	Iterative development Use case modeling Demonstration-based review
7. Analysis paralysis	Schedule	Demonstration-based review Use-case-oriented requirements/testing
8. Inadequate performance	Quality	Demonstration-based performance assessment Early architecture performance feedback
9. Overemphasis on artifacts	Schedule	Demonstration-based assessment Objective quality control
10. Inadequate function	Quality	Iterative development Early prototypes, incremental releases

Transitioning to a Modern Process

- Modern process features –
  - Early development of an initial version.
  - High risk areas are addressed early.
  - Several iterations are developed (called spirals, increments, generations, releases).
- Modern process characteristics-
  - Extensive use of domain experience.
  - Process flexibility and change management.
  - Architecture risk resolution.
  - Team cohesion.
  - Software process maturity.

Summary – The Old Way and the New Way

- Many “old way” principles still apply – make quality #1, determine the problem before writing the requirements, understand the customer’s priorities, etc…
- The “new way” principles feature architecture-first approaches, iterative development, and integrated change management.
- Transition from the “old” to the “new” requires early initial versions, addressing high risk areas early, evolving and refining the requirements, design and production, and involvement of the customer.

Assignment for Next Class

- Read Chapter 4 of Royce' book, on software development process.
  - Study the 30 principles of **conventional** software management.
  - Learn Royce' top 5 principles of **modern** software management
  - Study Royce' remaining top 10 principles for modern management.
- Read Chapter 4, "Aristocracy, Democracy, and System Design" of Brooks' book.
- If assigned to you, prepare the "Brooks' Chapter 4" 20 minute report (for presentation to the class).