The Source Code Control System
(Marc J. Rochkind: 1975)

- **Real-world problem:** Managing source code change:
  - Storage size for multiple production and development versions
  - Applying fixes to versions consistently
  - Determining exactly what changed, when, and why
  - Determining what version of the software customers have
- **Research problem:** Several Bell Labs software efforts
  - Largest with 100 programmers working on 2,964 code modules
- **Experiment:**
  - Examine problem elements and develop a “radical solution”
  - Build prototype (SCCS) and test on several real Bell Labs software development efforts—refine the solution via experience
- **Result:** SCCS very useful to programmers and management
- **Impact:** Introduced fundamentals of modern revision control
SCCS

- **Problem type:** Method/Improvement
  - What is a better way to do X?
- **Research model:** Minimize difference between practical and research setting by using SCCS prototype on actual Bell Labs software development
- **Hypothesis:** Source code management based upon storing programmer defined versions with efficient storage, protection, identification, and documentation is better than today’s (1975) practices
- **Strategy:** Describe concepts and design of SCCS prototype
- **Validation:**
  - Implementation (Successful IBM 370 & PDP-11 SCCS)
  - Experience (actual use at Bell Labs)

RCS—A System for Version Control
(Walter F. Tichy:1985)

- **Real-world problem:** Managing source code change:
  - Support multiple production and development versions
  - Controlling concurrent work on the same module of source code
  - Configuration management
- **Research problem:** Research software efforts at Purdue CSD
  - Examine current version control systems, examine problems, build RCS and use at Purdue CSD
- **Result:** RCS successful at Purdue CSD (Branching, locking, and better support for configuration management is useful)
- **Impact:** RCS added “practical” richness to SCCS model and provided most of the functionality important in a modern source code version control and configuration management system
• Problem type: Method/Improvement
  – What is a better way to do X?
• Research model:
  – Survey existing tools/ideas
  – Develop RCS prototype/production system at Purdue and use on research software development projects
• Hypothesis: Adding delta branching, check-out locking, and name/state/date based selection (configuration management) will improve upon the state-of-the-art (SCCS)
• Strategy: Implement RCS, describe how it works, refine with use on Purdue CSD projects
• Validation: Implementation (RCS prototype)