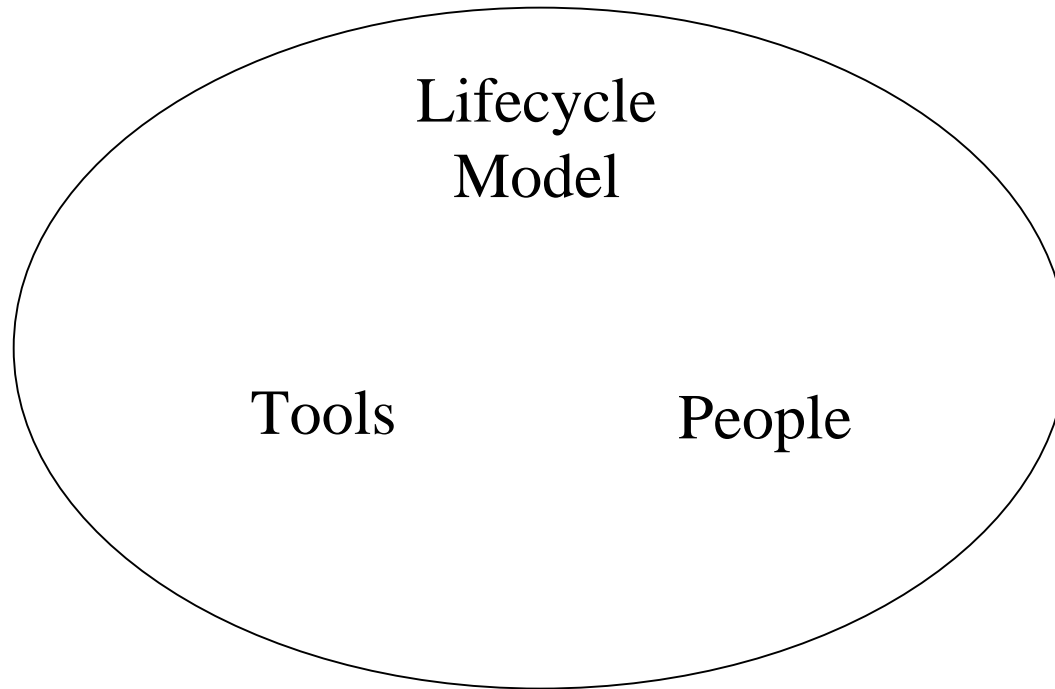


# The Software Process



# **If we don't govern ourselves, government will do it for us**

- 1987 – U.S. Department of Defense (DoD):  
“After two decades of largely unfulfilled promises about productivity and quality gains from applying new software methodologies and technologies, industry and government organizations are realizing that their fundamental problem is the inability to manage the software process.”

[“Report of the Defense Science Board Task Force on Military Software,” Office of Undersecretary of Defense for Acquisition, Washington, DC, September 1987.]

# Software Engineering Institute

- Founded **Software Engineering Institute (SEI)** at Carnegie Mellon University in Pittsburgh, Penn. SEI introduced
  - Series of tapes to teach software process, and a Master in Software Engineering
  - SEI Educator Development Workshop
  - **Capability Maturity Model (CMM)**
    - Lifecycle model independent strategies to improve software process
    - Five levels of maturity

# Capability Maturity Model (CMM)

## Five Models Adopted

SW-CMM – for software

P-CMM – for management of human resources (“people”)

SE-CMM – for systems engineering

IPS-CMM – for integrated product development

SA-CMM – for software acquisition

## Under Development...

??-CMM – for unifying the five models above – NEW INITIATIVE

# SW-CMM

- Proposed in 1986 by Watts Humphrey  
(Developer of Personal Software Process PSP)
- Underlying principle: The use of software techniques will not in itself result in increased productivity and profitability, because the problem is rooted in how we manage the software process.
- Change and improvements to the software process are gradual.
- An organization advances through **five levels of maturity**.

# **CMM – Five Levels of Maturity**

- **Maturity Level 1: Initial Level**
- **Maturity Level 2: Repeatable Level**
- **Maturity Level 3: Defined Level**
- **Maturity Level 4: Managed Level**
- **Maturity Level 5: Optimizing Level**

# CMM – Five Levels of Maturity

- **Maturity Level 1: Initial Level**

- No sound software engineering management practices in place.
- Everything is done on an adhoc basis.
- Usual patterns is time and cost overruns caused by a lack of sound management in general and planning in particular.
- Most activities are responses to crises, rather than preplanned tasks.
- Software process is unpredictable – dependent on current staff; if staff changes, so does the process.
- Impossible to predict with any accuracy such things as the time it will take to develop a product or the cost of that product.