

# Motivation

## ⇒ Micro-Evolution

↳ Evolve our understanding

➤ Requirements, architecture, design, code, test

↳ Evolve our documents and artifacts

## ⇒ Macro-Evolution

↳ Uses and features evolve over time

↳ Interdependencies among versions of systems

## ⇒ Persistence of artifacts

↳ Recreate arbitrary points in history of system

# Complexity of Artifacts

## ⇒ Compositional complexity

↳ Multiple parts that have to fit together

## ⇒ Temporal complexity

↳ Partial and incomplete compositions

## ⇒ Process complexity

↳ Private versus public availability

➤ Private: can do anything

➤ Public: have to worry about effects, implications

↳ Cooperation and coordination

➤ People

➤ projects

➤ Organizations

## ⇒ Evolution complexity

↳ Any change introduces a new version

↳ Worry about relationship among changes and versions

↳ For practical reasons, focus on logical sets of changes, not every individual change

# Solution: Configuration Mgmt

- ⇒ **Persistence of committed parts**
  - ↳ Persistence of all inconsequential, intermediate versions not practical
- ⇒ **Disciplined change**
  - ↳ Change Control Board oversight
  - ↳ Strict process of change
  - ↳ Request, approve, check-out, change, sign off, check-in
- ⇒ **Accountability**
  - ↳ Every change accounted for: who why, what, when
  - ↳ Versions, embody how
- ⇒ **All artifacts/documents**
  - ↳ Will focus on most common artifact: code