**Materialize process knowledge so it can...**
- Be easily passed on
- Guide process execution
- Assist in process evolution
- Enable better estimates

**Gave example in APPL/A language**
- "Most influential paper of ICSE 9"

**Followed Mary’s “new assumption -> new approach -> persuasion = sometimes a breakthrough (but sometimes nonsense) approach**

**Created field of process programming**

**Redwine/Riddle status:**
- Process programming made it to Internal Exploration
- RDL research largely backed up to Development and Extension
- Specific process definitions (TSP for example) in Popularization

**Process programming failed because...**
- Osterweil: APPL/A lacked appropriate constructs -> create JIL
- Others: it failed to take into account...
  - Evolution -> SPADE (rigidly), APEL, Sentinel, PROSYT
  - Teams of teams:
    - Decentralized environments -> Oz, PROSYT
    - Heterogeneity of PSEEs -> APEL, Oz
  - Deviations -> Endeavors, PROSYT
  - Artifacts as common denominator -> PROSYT, CMU
  - Bugzilla/CVS research project

**Key remaining questions**
- Best balance?
  - Cooperation Vs automation
  - Flexibility Vs control
  - Rich process data Vs minimal intrusion
- How to deal with...?
  - Deviations between definition and the process actually followed
  - Distributed teams
  - Nomadic workers
- Is artifact orientation better than process orientation?
- Can we leverage the progress that the business process folks have achieved with concepts like Petri Nets, Pi Calculus and the various BPMLs?