Group One
• Conventional software tools do not cover the embedded system domain
  – Distributed Scheduling, Clock Drift, Failures
  – Memory and other resource limits
• Integrating Point Solutions
  – Consistency with neighboring processes
  – Vetting assumptions
  – Tool interoperability, ontologies, translations
• Documentation
  – Transforming human views to tool views and vice versa

• Legacy systems
  – Reengineering and adaptation
  – Domain specific languages, match domain terminology

• Tools need re-thinking - end to end problem
  – New kinds of tools
  – Aid for domain specific language design
  – Tailoring to process and domain expertise
  – Methods for requirement analysis when many inter-related issues can not be separated