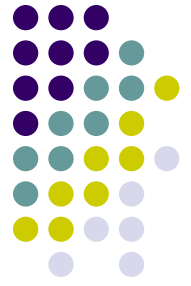


Requirements Engineering as a Failed Discipline



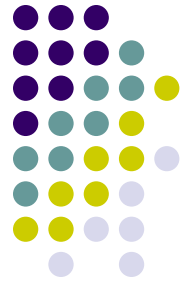
Richard N. Taylor
Institute for Software Research
University of California, Irvine
<http://www.ics.uci.edu/~taylor>



Why Are Requirements Done So Poorly, After the Fact, or If at All, in So Many Applications?

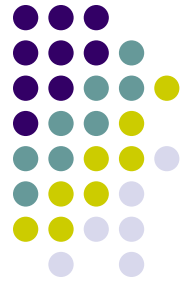
- Standard answers:
 - Bad engineering
 - Bad discipline
 - Lack of good mathematical training
 - Lack of time

Maybe the Reason Is Different



- Maybe it is because it hasn't proven useful
- Maybe it is because you can't do a good job with requirements until the architecture is in hand
- Maybe it is a matter of size and complexity
- Maybe Petroski is right: failure is the driver of engineering and the basis for innovation

Alternative: Architectures in the Lead



- Think of requirements as incremental improvements needed to existing architectures, or as compositions of architectures
- Architectures provide a frame of reference
 - a vocabulary
 - a basis for describing properties
 - a basis for analysis
- Create new architectures based upon experience with and improvement to pre-existing architectures

**Are All Architectures up to the Task
of Being “Improved” in a Cost-effective Way?**

Do We Need Requirements at all?



- You do have to know your objective before you start new work.
- You do need a contract with the customer
 - (but when you are building to a market?)
- **But let (substantive) architectures:**
 - Provide the vocabulary
 - Provide the basis for discussion
 - ... as well as *being* the solution basis
- Thus: new objectives and solutions, from old problems and old solutions