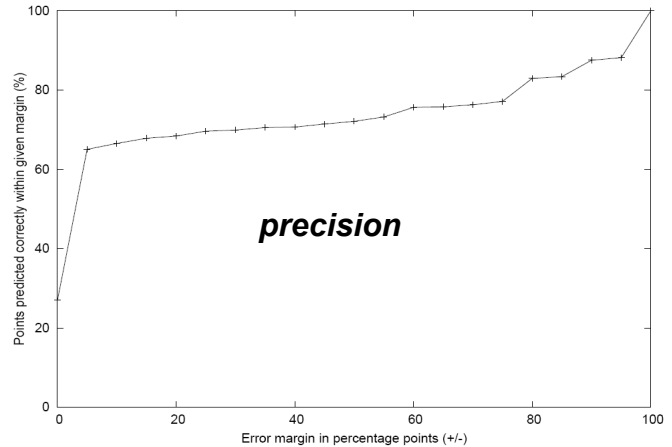


# Prioritizing Software Inspection Results using Execution Likelihood



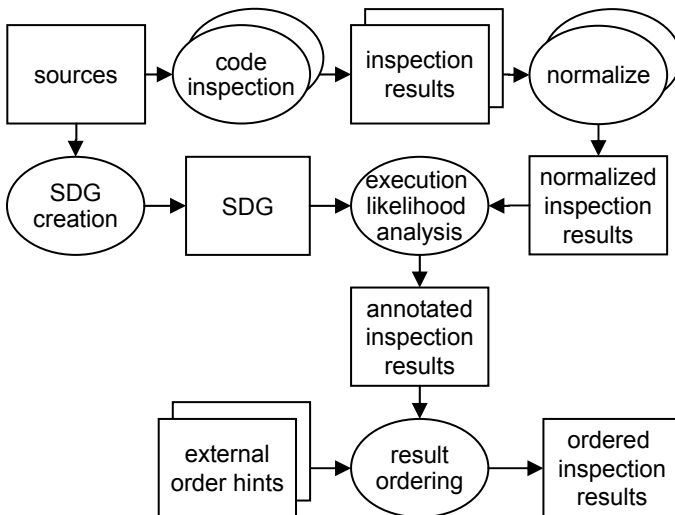
## Why Prioritize Inspection Results?

- Automated software inspection tools often produce an overly large number of warnings, thereby effectively limiting their usefulness
- Frequent question: "What are good starting points for processing the list of reported issues?"
- Observation: Only execution will trigger a fault
- Idea: provide a prioritization based on the likelihood that locations are executed
- Allows to focus on most pertinent issues first



## ELAN at a glance

- ELAN: Execution Likelihood Analysis
- Based on graph representation of the program's source code (SDG)
- Reachability analysis (program slicing) limits graph portion to be investigated
- Simple traversal determines all valid acyclic execution paths
- Result is a *demand-driven, static, program analysis*
- Approach was evaluated using a benchmark suite of programs up to 67KLoC (industrial) C code



## ELAN strengths

- Can be used in conjunction with any software inspection tool
- Complementary to existing prioritization techniques, such as Z-ranking
- Using simple mechanics, reasonable accurate predictions can be obtained
- Performance scales well to large industrial systems

