SE-3002
SOFTWARE QUALITY ENGINEERING

RUBAB JAFFAR
RUBAB.JAFFAR@NU.EDU.PK

Part IV-QUANTIFIABLE QUALITY IMPROVEMENT Feedback Loop and Activities for Quantifiable Quality Improvement

Lecture # 36

9 Dec

TODAY'S OUTLINE

- Feedback Loop and Activities for Quantifiable Quality Improvement
- Feedback Loop and Overall Mechanism
- Monitoring and Measurement
- Analysis and Feedback
- Tool and Implementation Support

QUANTIFIABLE QUALITY IMPROVEMENT BASIC ELEMENTS

- Part IV is quantifiable quality improvement, which includes two basic elements:
- Quantification of quality through quantitative measurements and models so that the quantified quality assessment results can be compared to the pre-set quality goals for quality and process management.
- Quality improvement through analyses and follow-up activities by identifying quality improvement possibilities, providing feedback, and initiating follow-up actions.

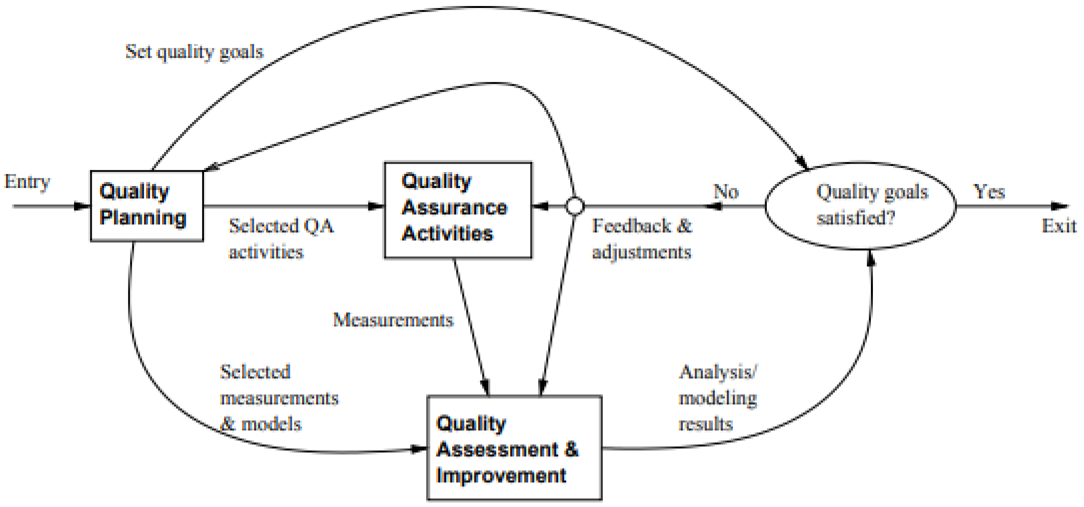
QUANTIFIABLE QUALITY IMPROVEMENT

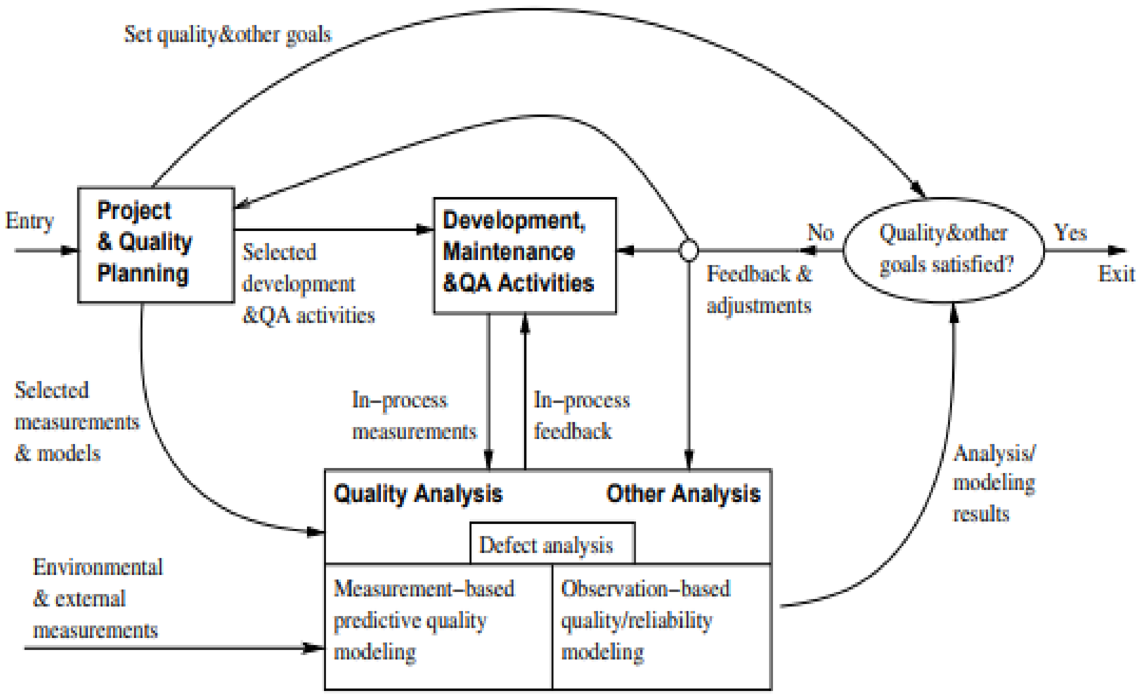
- To support quantifiable quality improvement, various parallel and follow-up activities to the main quality assurance (QA) activities are needed, including:
- Monitoring the specific QA activities and the overall software development or maintenance activities, and extracting relevant measurement data.
- Analyzing the data collected above for quality quantification and identification of quality improvement opportunities.
- Providing feedback to the QA and development/maintenance activities and carrying out follow-up actions based on the analysis results above.

QUANTIFIABLE QUALITY IMPROVEMENT

- These activities also close the quality engineering feedback loop discussed in part I and refine it into Following:

Software quality engineering (SQE)





IMPORTANCE OF FEEDBACK LOOP

- All QA activities covered in Part II and Part III need additional support: .
- Planning and goal setting
- Management via feedback loop
  - When to stop?
  - Adjustment and improvement, etc.
  - All based on assessments/predictions

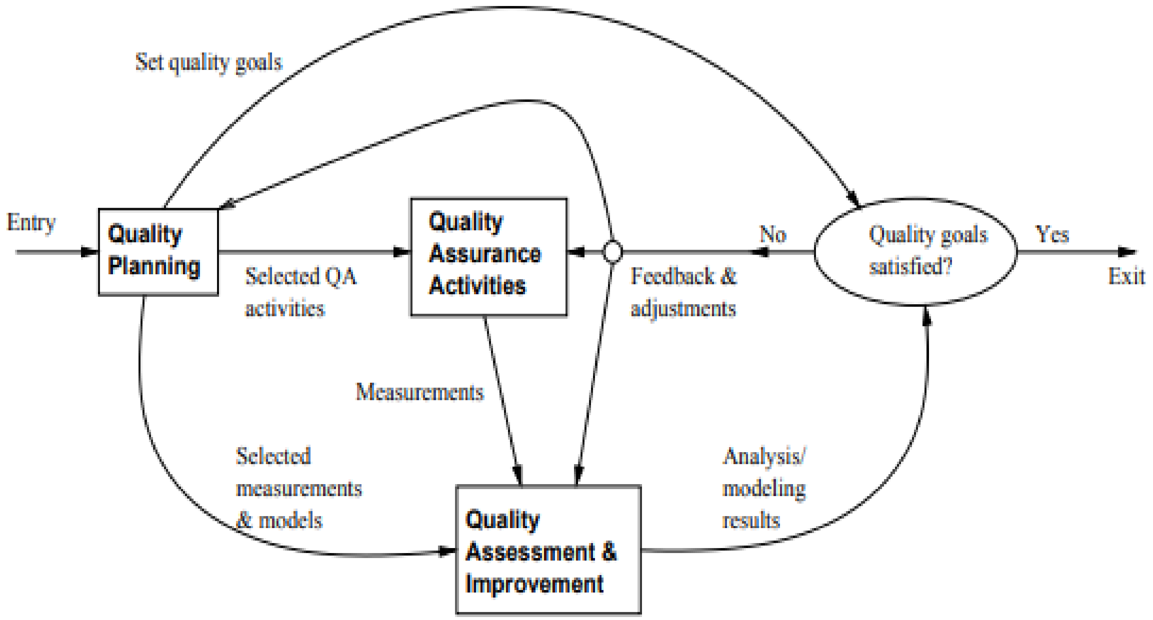
IMPORTANCE OF FEEDBACK LOOP

- Feedback loop for quantification/improvement: i.e. Focus of Part IV
  - mechanism and implementation
  - models and measurements.
  - defect analyses and techniques
  - risk identification techniques
  - software reliability engineering

QE ACTIVITIES AND PROCESS REVIEW

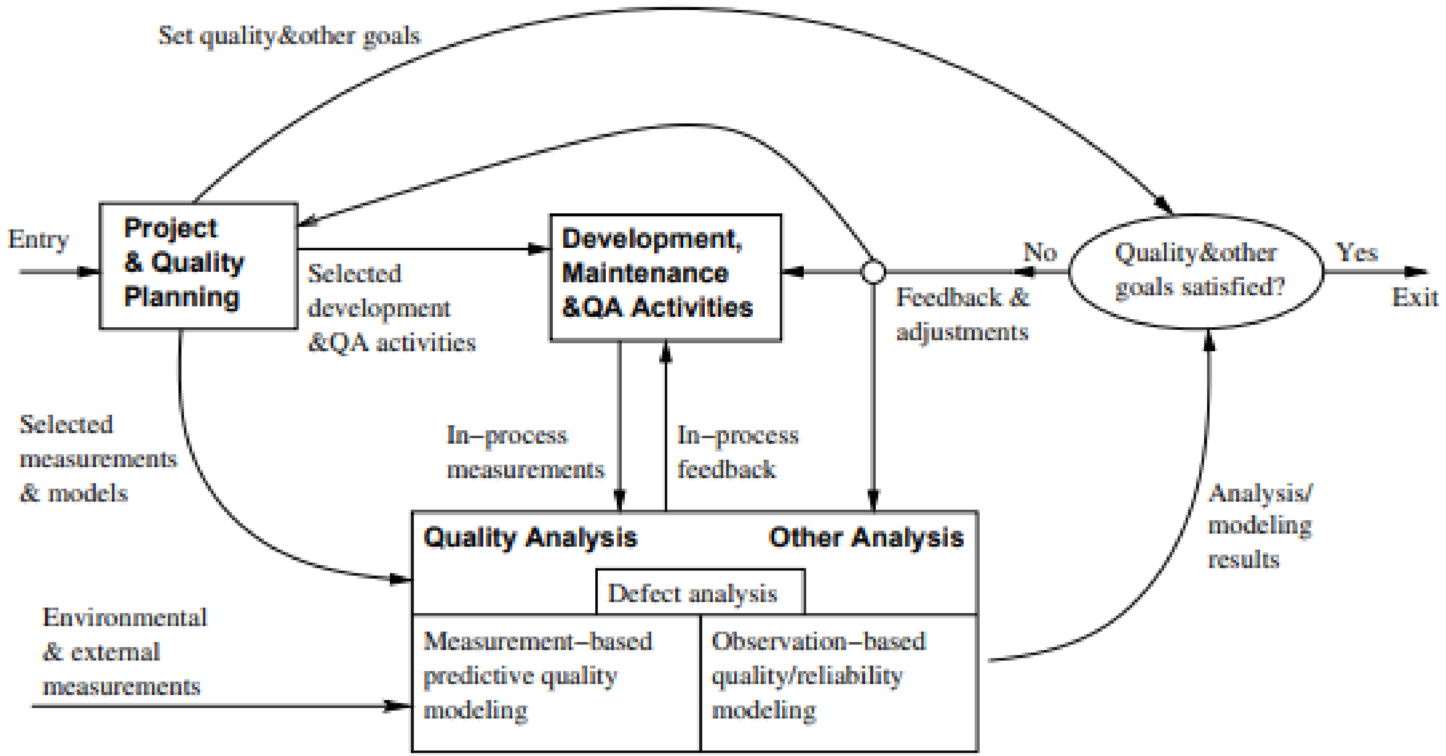
Software quality engineering (SQE)

- Major activities:
- Pre-QA planning (Part I)
- QA (Part II and Part III)
- Post-QA analysis & feedback – Part IV (maybe parallel instead of "post-")



QE ACTIVITIES AND PROCESS REVIEW

- Feedback loop zoom-in:
- Multiple measurement sources.
- Many types of analysis performed.
- Multiple feedback paths.



FEEDBACK LOOP RELATED ACTIVITIES

- Monitoring and measurement: .
  - defect monitoring ∈ process management
  - defect measurement ∈ defect handling
  - many other related measurements.
- Analysis modeling:
  - Historical baselines and experience
  - Choosing models and analysis techniques
  - Focus on defect/risk/reliability analyses
  - Goal: assessment/prediction/improvement.
- Feedback and follow-up
  - Frequent feedback: assessment/prediction
  - Possible improvement areas identified
  - Overall management and improvement.

QUALITY MONITORING AND MEASUREMENTS

- Quality monitoring needs:
  - Quality as a quantified entity over time
  - Able to assess, predict, and control
  - Various measurement data needed
- Some directly in quality monitoring
- Others via analyses to provide feedback.
- Direct quality measurements:
  - Result, impact and related info. e.g., success vs. failure
- Defect information: directly monitored.
- Mostly used in quality monitoring.

INDIRECT QUALITY MEASUREMENTS

- Indirect quality measurements: Why?
- Other quality measurements (reliability) need additional analyses/data.
- Unavailability of direct quality measurements early in the development cycle ⇒ early (indirect) indicators.
- Used to assess/predict/control quality. (to link to or affect various direct quality measurements)
- Types of indirect quality measurements:
  - Environmental measurements.
  - Product internal measurements.
  - Activity measurements.

INDIRECT MEASUREMENTS: ENVIRONMENT

- Process characteristics
  - Entities and relationships
  - Preparation, execution and follow-up
  - Techniques used
- People characteristics
  - Skills and experience
  - Roles: planners/developers/testers
  - Process management and teams
- Product characteristics
  - Product/market environment
  - Hardware/software environment

INDIRECT MEASUREMENTS: INTERNAL

- Product internal measurements: most studied/understood in SE
- Software artifacts being measured:
  - Mostly code-related
  - Sometimes SRS, design, docs etc.
- Product attributes being measured:
  - Control: e.g., McCabe complexity
  - Data: e.g., Halstead metrics
  - Presentation: e.g., indentation rules

INDIRECT MEASUREMENTS: ACTIVITY

- Execution/activity measurements:
  - Overall: e.g., cycle time, total effort.
  - Phased: profiles/histograms.
  - Detailed: transactions
- Testing activity examples:
  - Timing during testing/usage
  - Path verification (white-box)
  - Usage-component mapping (black-box)
  - Measurement along the path
- Usage of observations/measurements: observation-based and predictive models

IMMEDIATE FOLLOW-UP AND FEEDBACK

- Immediate (without analyses): Why?
  - Immediate action needed right away:
    - critical problems $\Rightarrow$ immediate fixing
    - most other problems: no need to wait
  - Some feedback as built-in features in various QA alternatives and techniques.
  - Activities related to immediate actions.
- Testing activity examples:
  - Shifting focus from failed runs/areas.
  - Re-test to verify defect fixing.
  - Other defect-related adjustments.
- Defect and activity measurements used.

ANALYSES, FEEDBACK, AND FOLLOW-UP

- Most feedback/followup relies on analyses.
- Types of analyses:
  - Product release decision related.
  - For other project management decisions, at the phase or overall project level.
  - Longer-term or wider-scope analyses.
- Types of feedback paths:
  - Shorter vs. longer feedback loops.
  - Frequency and time duration variations.
  - Overall scope of the feedback.
  - Data source refinement.
  - Feedback destinations.

ANALYSIS FOR PRODUCT RELEASE DECISIONS

- Most important usage of analysis results
  - Prominent in SQE and modified SQE Figure
  - Related to: "when to stop testing?"
- Basis for decision making:
  - Without explicit quality assessment:
    - implicit: planned activities,
    - indirect: coverage goals,
    - other factors: time/$-based.
  - With explicit quality assessment:
    - failure-based: reliability,
    - fault-based: defect count & density.
- Criteria preference:
  - reliability – defect – coverage – activity.

ANALYSES FOR OTHER DECISIONS

- Transition from one (sub-)phase to another:
  - Later ones: similar to product release.
  - Earlier ones: reliability undefined
    - defects – coverage – activity,
    - inspection and other early QA
- Other decisions/management-activities:
  - Schedule adjustment.
  - Resource allocation and adjustment.
  - Planning for post-release support.
  - Planning for future products or updates.
- These are product-level or sub-product-level decisions and activities

OTHER FEEDBACK AND FOLLOWUP

- Other (less frequent) feedback/followup:
  - Goal adjustment (justified/approved).
  - Self-feedback (measurement & analysis)
  - Longer term, project-level feedback.
  - May even carry over to followup projects.
- Beyond a single-project duration/scope:
  - Future product quality improvement
    - overall goal/strategy/model/data,
    - especially for defect prevention.
- Process improvement.
- More experienced people

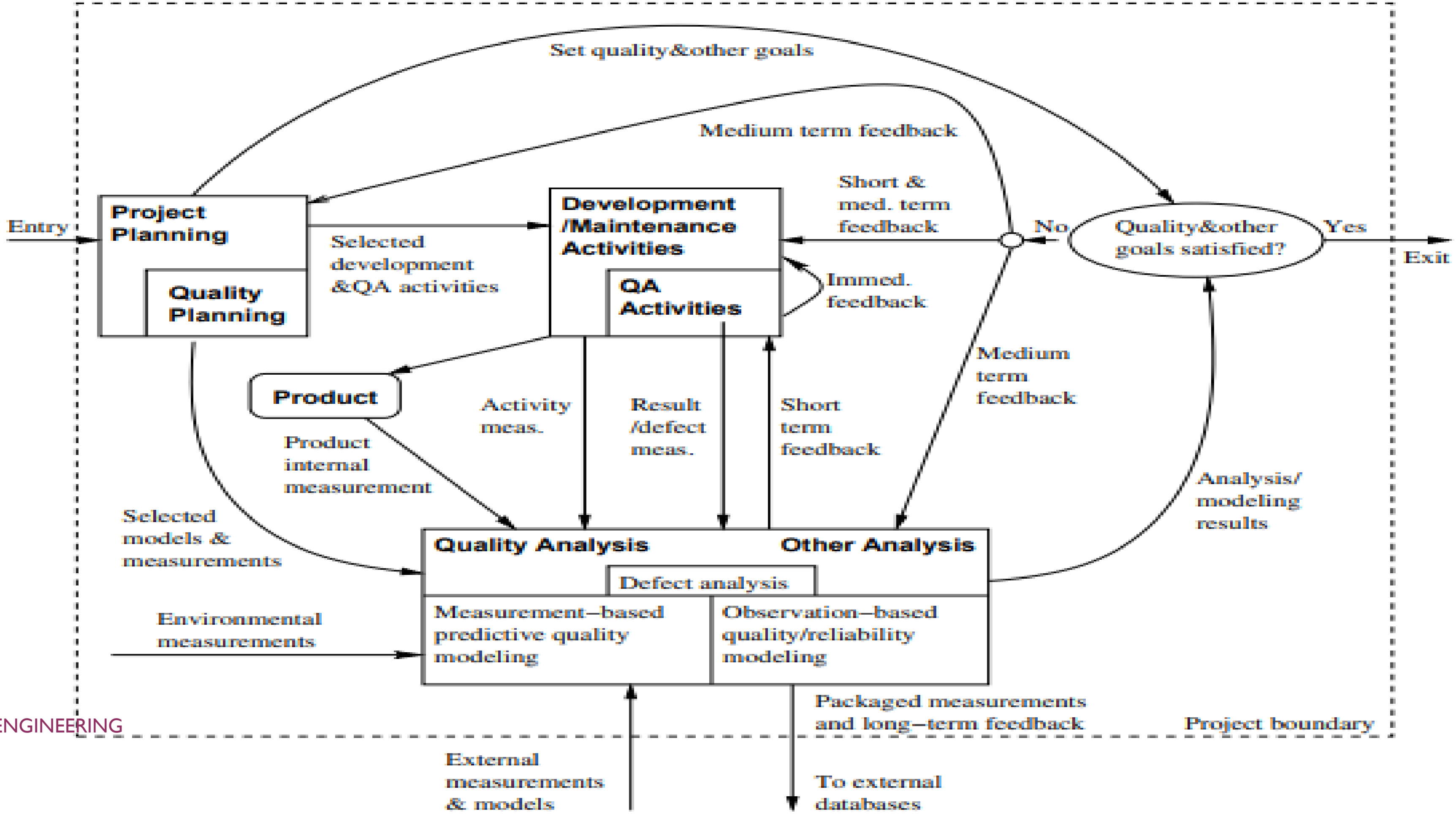
FEEDBACK LOOP IMPLEMENTATION

- Key question: sources and destinations. (Analysis and modeling activity at center.)
- Sources of feedback loop = data sources:
  - Result and defect data:
    - the QA activities themselves.
  - Activity data:
    - both QA and development activities.
  - Product internal data: product. (produced by development activities)
  - Environmental data: environment.
- Additional sources of feedback loop:
  - From project/QA planning.
  - Extended environment: measurement data and models beyond project scope.

FEEDBACK LOOP IMPLEMENTATION

- Feedback loop at different duration/scope levels.
- Immediate feedback to current development activities (locally).
- Short-term or sub-project-level feedback:
  - most of the feedback/followup
  - transition, schedule, resource,
  - destination: development activities.
- Medium-term or project-level feedback:
  - overall project adjustment and release
  - destination: major blocks
- Longer-term or multi-project feedback:
  - to external destinations

FEEDBACK LOOP IMPLEMENTATION



IMPLEMENTATION SUPPORT TOOLS

- Type of tools:
  - Data gathering tools.
  - Analysis and modeling tools.
  - Presentation tools.
- Data gathering tools:
  - Defects/direct quality measurements:
    - from defect tracking tools.
  - Environmental data: project db.
  - Activity measurements: logs.
  - Product internal measurements:
    - commercial/home-build tools.
  - New tools/APIs might be needed.

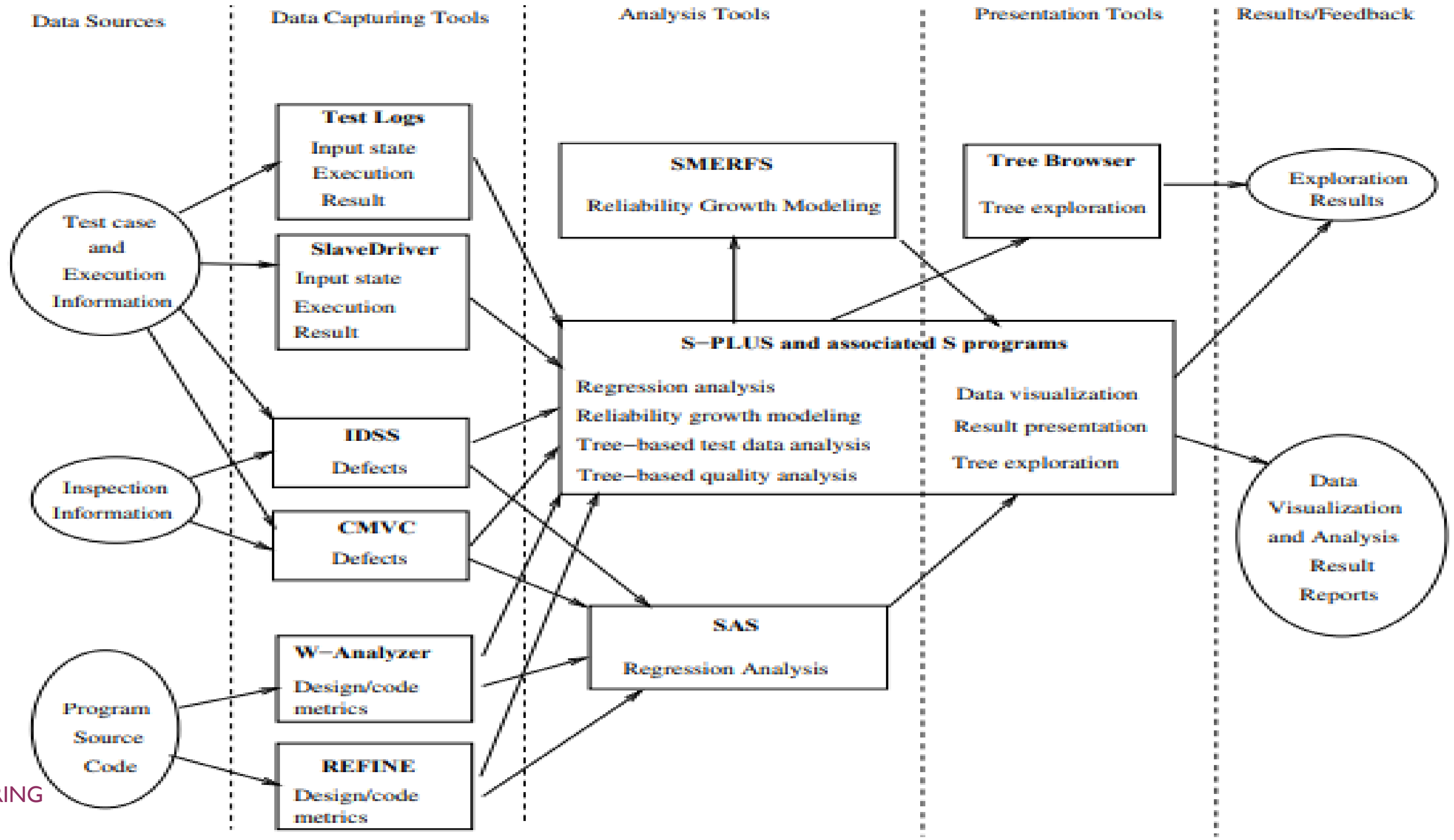
IMPLEMENTATION SUPPORT TOOLS

- Analysis and modeling tools:
  - Dedicated modeling tools:
    - e.g., SMERFS and CASRE for SRE
  - General modeling tools/packages:
    - e.g., multi-purpose S-Plus, SAS.
  - Utility programs often needed for data screening and processing.
- Presentation tools:
  - Aim: easy interpretation of feedback ⇒ more likely to act on.
  - Graphical presentation preferred.
  - Some "what-if"/exploration capability.

STRATEGY FOR TOOL SUPPORT

- Using existing tools $\Rightarrow$ cost$\downarrow$:
  - Functionality and availability/cost.
  - Usability.
  - Flexibility and programmability.
  - Integration with other tools.
- Tool integration issues:
  - Assumption: multiple tools used. (All-purpose tools not feasible/practical.)
  - External rules for inter-operability,
    - common data format and repository.
  - Multi-purpose tools.
  - Utilities for inter-operability.

TOOL SUPPORT EXAMPLE



That is all