Software Testing Maturity Model^{SM} (SW-TMM^{SM})

Presenter: Duy Huynh

- **Thomas C. Staab. Using SW-TMM to Improve the Testing Process. Crosstalk The Journal of Defense Software Engineering. November 2002.**
- Ilene Burnstein, Ariya Homyen, Robert Grom, and C.R. Carlson. A Model to Access Testing Process Maturity. Crosstalk The Journal of Defense Software Engineering. November 1998.
- Ilene Burnstein, Taratip Suwannasart, and C.R. Carlson. Developing a Testing Maturity Model: Part I. Crosstalk The Journal of Defense Software Engineering. August 1996.
- Ilene Burnstein, Taratip Suwannasart, and C.R. Carlson. Developing a Testing Maturity Model: Part II. Crosstalk The Journal of Defense Software Engineering. September 1996.
- The Capability Maturity Model at Software Engineering Institute www.sei.cmu.edu.
- Personal contact with Thomas C. Staab.

Outline

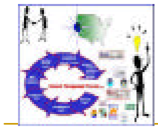
- Part I: Capability Maturity Model
  - Fundamental concepts underlying process maturity
  - The Software Capability Maturity Model (SW-CMM®)
- Part II: Software Testing Maturity Model
  - History
  - Fundamental concepts underlying SW-TMM
  - Why do we need SW-TMM?
  - How do we perform SW-TMM assessment?

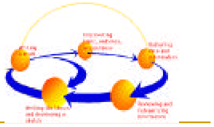
Part I

- Part I: Capability Maturity Model
  - Fundamental concepts underlying process maturity
  - The Software Capability Maturity Model (SW-CMM®)
- Part II: Software Testing Maturity Model
  - History
  - Fundamental concepts underlying SW-TMM
  - Why do we need SW-TMM?
  - How do we perform SW-TMM assessment?

Software Process

- A software process can be defined as a set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products





Software Process Capability

- Software Process Capability describes the range of expected results that can be achieved by following a software process.
- The software process capability of an organization provides one means of predicting the most likely outcomes to be expected from the next software project the organization undertakes

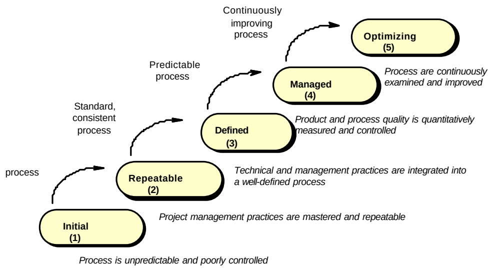
Software Process Maturity

- Software Process Maturity is the extent to which a specific process is explicitly defined, managed, measured, controlled, and effective.
- Maturity implies a potential growth in capability and indicates both the richness of an organization's software process and the consistency with which it is applied in projects throughout the organization

Why is software process maturity important?

- Increase the long-term success of a project and the long-term health of an organization
- Stabilize the development lifecycle
- Reduce software development risk
- Focus creativity on the right areas
- Repeatability
- Predictability
- Manageability
- Etc

The five levels of SW-CMM


Continuously improving process
Optimizing (5)
Predictable process
Managed (4)
Process are continuously examined and improved
Standard, consistent process
Defined (3)
Product and process quality is quantitatively measured and controlled
process
Repeatable (2)
Technical and management practices are integrated into a well-defined process
Initial (1)
Project management practices are mastered and repeatable
Process is unpredictable and poorly controlled

Other models being developed by SEI

- P-CMM: People Capability Maturity Model
- SA-CMM: Software Acquisition Capability Maturity Model
- SE-CMM: Systems Engineering Capability Maturity Model
- IPD-CMM: Integrated Product Development Capability Maturity Model
- CMMI®: Capability Maturity Model Integration

Outline

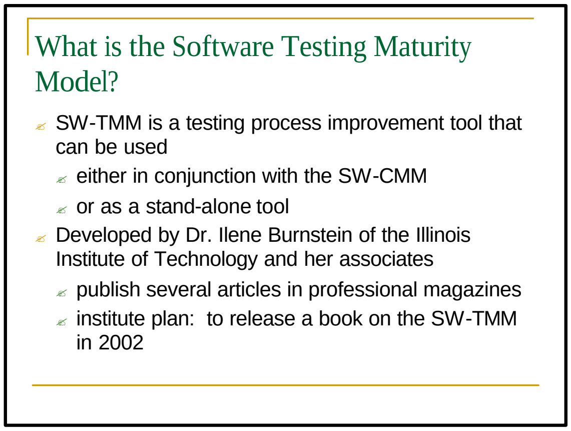
- Part I: Capability Maturity Model
  - Fundamental concepts underlying process maturity
  - The Software Capability Maturity Model (SW-CMM®)
- Part II: Software Testing Maturity Model
  - History of testing maturity models
  - Fundamental concepts underlying SW-TMM
  - Why do we need to use SW-TMM?
  - How do we perform SW-TMM assessment?

Why should we assess the testing maturity?

- There is no consistency within their organization as to the health and professionalism of the testing process
- An assessment of the testing process using a testing maturity model will
  - document the current level
  - highlight the variances between the imagined level and the actual level
  - provide a road map for making the necessary process improvements

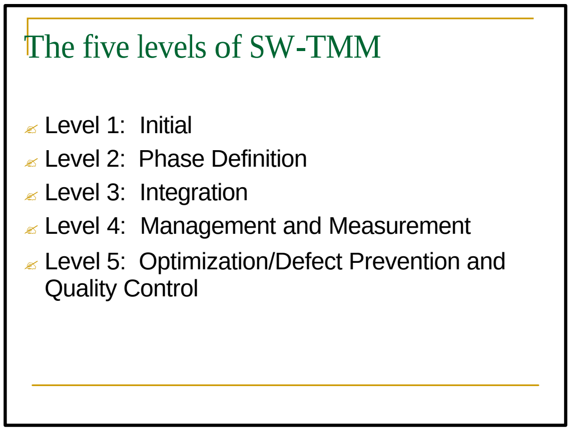
Testing maturity models

- Most of the testing maturity models were developed around 1996
  - Testability Maturity Model
  - Software Testing Maturity Model
  - Test Process Improvement
  - Test Organization Maturity™
  - Testing Assessment Program
  - Proposed Evaluation and Test SW-CMM Key Process Area
- None has found much acceptance because of
  - little documentation on the model
  - theoretical style



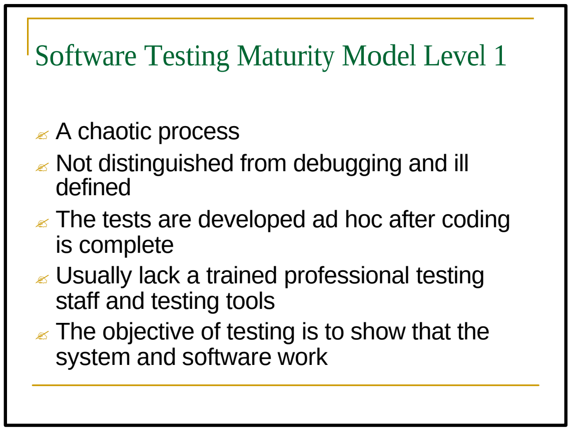
What is the Software Testing Maturity Model?

- SW-TMM is a testing process improvement tool that can be used
  - either in conjunction with the SW-CMM
  - or as a stand-alone tool
- Developed by Dr. Ilene Burnstein of the Illinois Institute of Technology and her associates
  - publish several articles in professional magazines
  - institute plan: to release a book on the SW-TMM in 2002



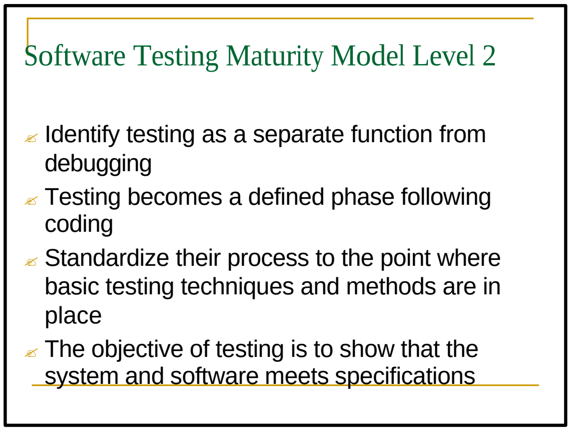
The five levels of SW-TMM

- Level 1: Initial
- Level 2: Phase Definition
- Level 3: Integration
- Level 4: Management and Measurement
- Level 5: Optimization/Defect Prevention and Quality Control



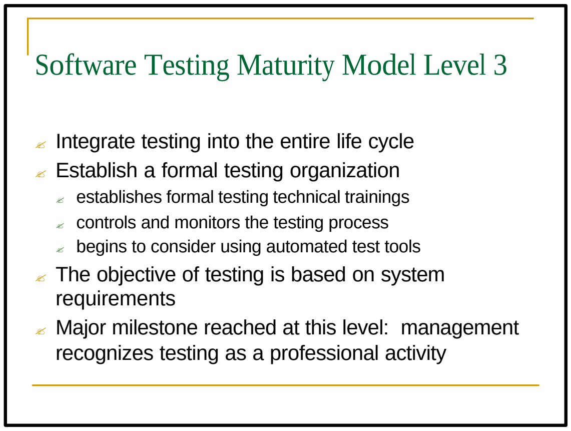
Software Testing Maturity Model Level 1

- A chaotic process
- Not distinguished from debugging and ill defined
- The tests are developed ad hoc after coding is complete
- Usually lack a trained professional testing staff and testing tools
- The objective of testing is to show that the system and software work



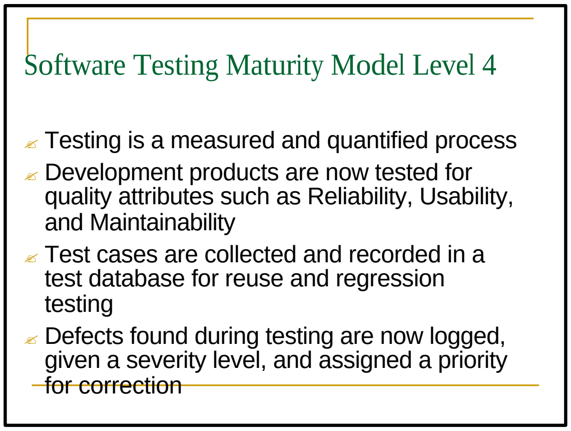
Software Testing Maturity Model Level 2

- Identify testing as a separate function from debugging
- Testing becomes a defined phase following coding
- Standardize their process to the point where basic testing techniques and methods are in place
- The objective of testing is to show that the system and software meets specifications



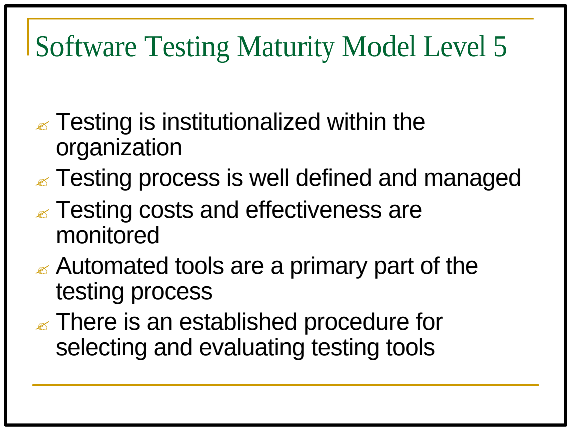
Software Testing Maturity Model Level 3

- Integrate testing into the entire life cycle
- Establish a formal testing organization
  - establishes formal testing technical trainings
  - controls and monitors the testing process
  - begins to consider using automated test tools
- The objective of testing is based on system requirements
- Major milestone reached at this level: management recognizes testing as a professional activity



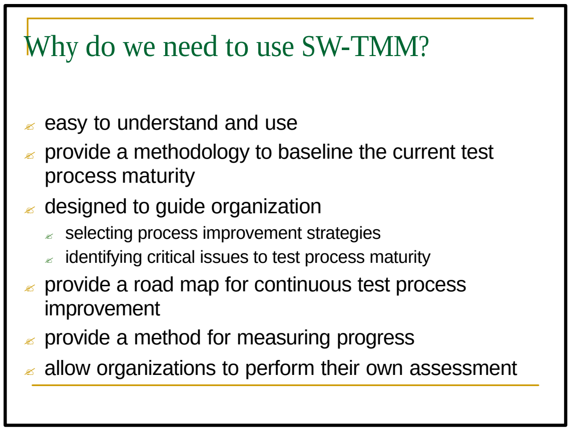
Software Testing Maturity Model Level 4

- Testing is a measured and quantified process
- Development products are now tested for quality attributes such as Reliability, Usability, and Maintainability
- Test cases are collected and recorded in a test database for reuse and regression testing
- Defects found during testing are now logged, given a severity level, and assigned a priority for correction



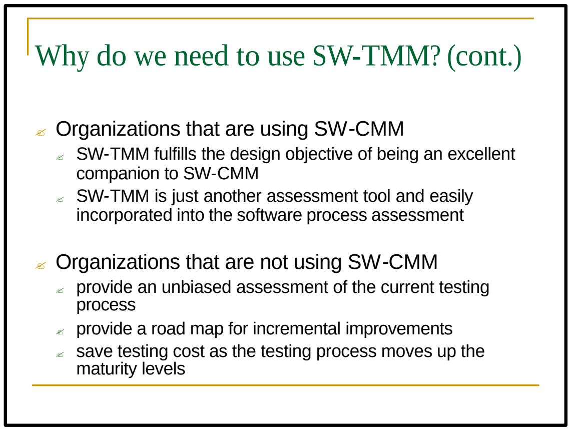
Software Testing Maturity Model Level 5

- Testing is institutionalized within the organization
- Testing process is well defined and managed
- Testing costs and effectiveness are monitored
- Automated tools are a primary part of the testing process
- There is an established procedure for selecting and evaluating testing tools



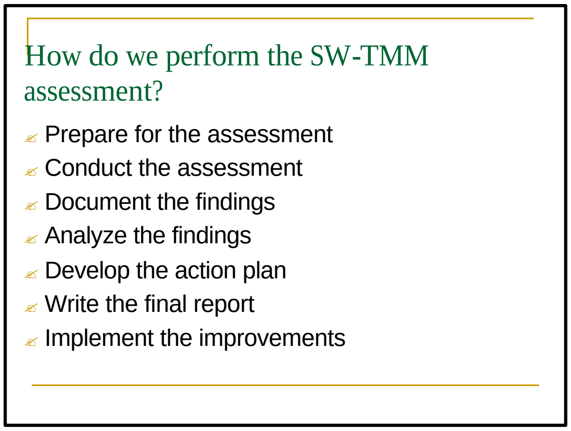
Why do we need to use SW-TMM?

- easy to understand and use
- provide a methodology to baseline the current test process maturity
- designed to guide organization
  - selecting process improvement strategies
  - identifying critical issues to test process maturity
- provide a road map for continuous test process improvement
- provide a method for measuring progress
- allow organizations to perform their own assessment



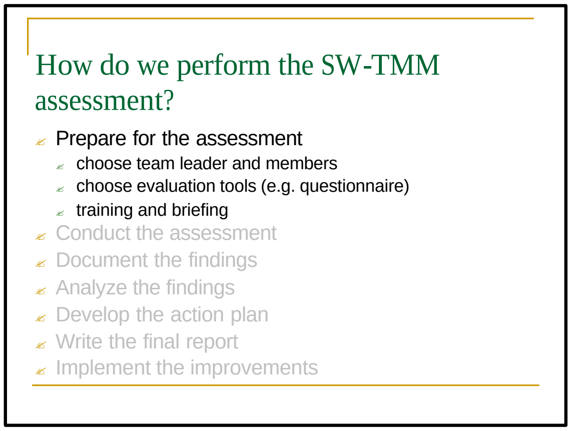
Why do we need to use SW-TMM? (cont.)

- Organizations that are using SW-CMM
  - SW-TMM fulfills the design objective of being an excellent companion to SW-CMM
  - SW-TMM is just another assessment tool and easily incorporated into the software process assessment
- Organizations that are not using SW-CMM
  - provide an unbiased assessment of the current testing process
  - provide a road map for incremental improvements
  - save testing cost as the testing process moves up the maturity levels



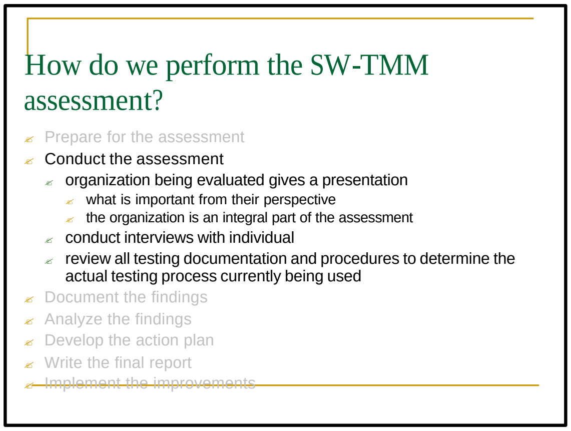
How do we perform the SW-TMM assessment?

- Prepare for the assessment
- Conduct the assessment
- Document the findings
- Analyze the findings
- Develop the action plan
- Write the final report
- Implement the improvements



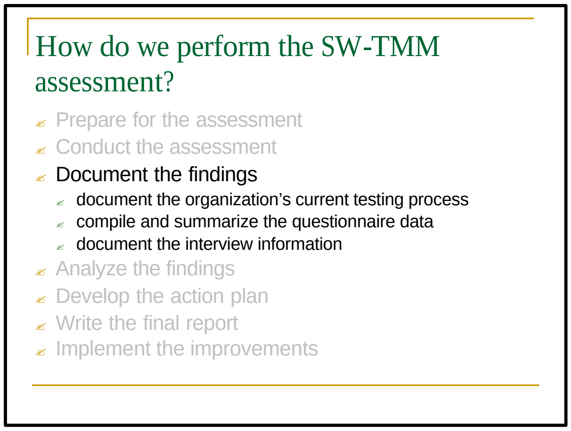
How do we perform the SW-TMM assessment?

- Prepare for the assessment
  - choose team leader and members
  - choose evaluation tools (e.g. questionnaire)
  - training and briefing
- Conduct the assessment
- Document the findings
- Analyze the findings
- Develop the action plan
- Write the final report
- Implement the improvements



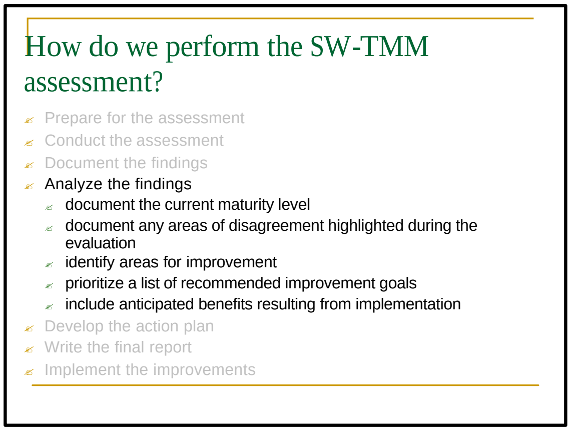
How do we perform the SW-TMM assessment?

- Prepare for the assessment
- Conduct the assessment
  - organization being evaluated gives a presentation
    - what is important from their perspective
    - the organization is an integral part of the assessment
  - conduct interviews with individual
  - review all testing documentation and procedures to determine the actual testing process currently being used
- Document the findings
- Analyze the findings
- Develop the action plan
- Write the final report
- Implement the improvements



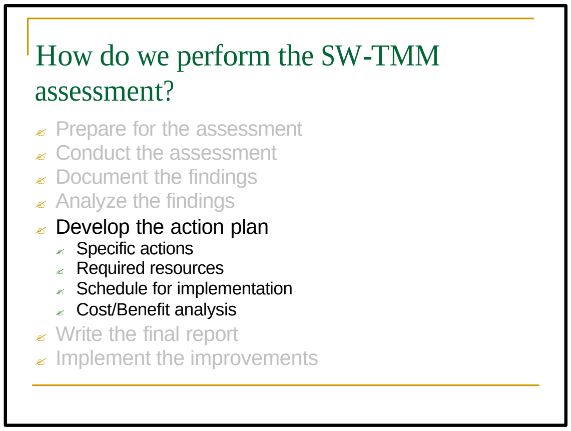
How do we perform the SW-TMM assessment?

- Prepare for the assessment
- Conduct the assessment
- Document the findings
  - document the organization’s current testing process
  - compile and summarize the questionnaire data
  - document the interview information
- Analyze the findings
- Develop the action plan
- Write the final report
- Implement the improvements



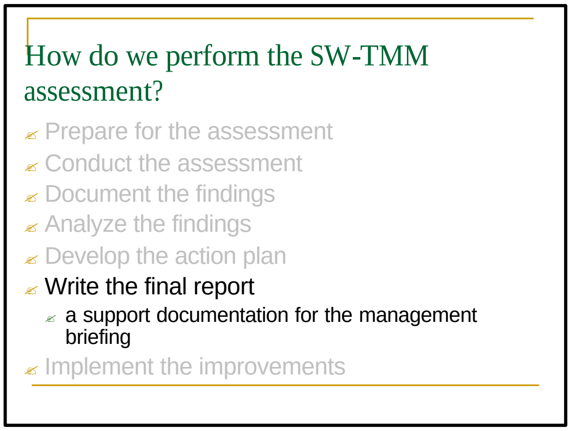
How do we perform the SW-TMM assessment?

- Prepare for the assessment
- Conduct the assessment
- Document the findings
- Analyze the findings
  - document the current maturity level
  - document any areas of disagreement highlighted during the evaluation
  - identify areas for improvement
  - prioritize a list of recommended improvement goals
  - include anticipated benefits resulting from implementation
- Develop the action plan
- Write the final report
- Implement the improvements



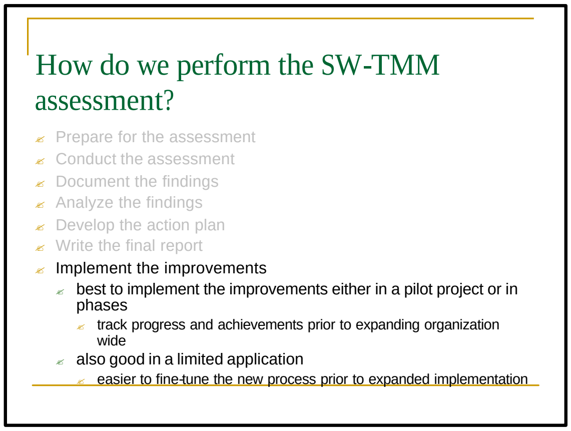
How do we perform the SW-TMM assessment?

- Prepare for the assessment
- Conduct the assessment
- Document the findings
- Analyze the findings
- Develop the action plan
  - Specific actions
  - Required resources
  - Schedule for implementation
  - Cost/Benefit analysis
- Write the final report
- Implement the improvements



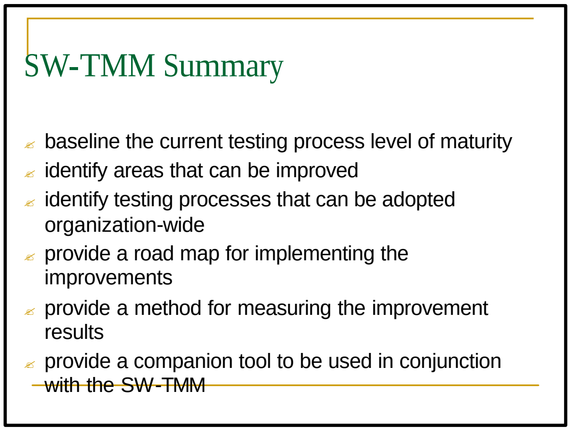
How do we perform the SW-TMM assessment?

- Prepare for the assessment
- Conduct the assessment
- Document the findings
- Analyze the findings
- Develop the action plan
- Write the final report
  - a support documentation for the management briefing
- Implement the improvements



How do we perform the SW-TMM assessment?

- Prepare for the assessment
- Conduct the assessment
- Document the findings
- Analyze the findings
- Develop the action plan
- Write the final report
- Implement the improvements
  - best to implement the improvements either in a pilot project or in phases
    - track progress and achievements prior to expanding organization wide
  - also good in a limited application
    - easier to fine-tune the new process prior to expanded implementation



SW-TMM Summary

- baseline the current testing process level of maturity
- identify areas that can be improved
- identify testing processes that can be adopted organization-wide
- provide a road map for implementing the improvements
- provide a method for measuring the improvement results
- provide a companion tool to be used in conjunction with the SW-TMM