מיסמכולוגיה

תקנים לפיתוח תוכנה

אוהד ברזילי

פיתוח מערכות תוכנה מבוססות Java
בית הספר למדעי המחשב, אוניברסיטת תל אביב

היום בשיעור

☐ למה תקנים?

1. ISO/IEC 90003 - a Quality Management Standard for software products and related services

2. ISO/IEC 12207 - Standard for Information Technology – Software Life Cycle Processes

3. CMMI - Capability Maturity Model Integration

תו תקן לפיתוח תוכנה

- "משבר התוכנה" הוא כינוי לאי עמידה בזמנים, אי עמידה בדרישות, אי עמידה באכויות ואי עמידה בתקציב והתמיכה הלקויה המאפיינים את עולם התוכנה

- כדי להתמודד עם האתגר נעשים מאמצים בכיוונים שונים ומהבטים שונים

- אחד מהם הוא שאיפה לפיתוח "תו תקן" לפיתוח תוכנה איכותי

תו תקן לפיתוח תוכנה

☐ בעולם כמה גופי תקינה המפרסמים תקנים
(מסמכים) המגדירים איכות פיתוח מהבטים שונים

☐ גופי תוכנה יכולים לקבל **הסמכה** כי הם תואמים
לתקן

☐ בחלק מהתחומים תקנים אלו הכרחיים, ויש לקוחות
המתנים את פיתוח התוכנה עבורם בעמידה בתקן
מסוים (צבא, רפואה, גופים ממלכתיים)

דיון

☐ לא כל התוכנות אותו הדבר

■ איך קובעים תקן שיתאים לכל סוגי התוכנה?

☐ וגם לא כל הגופים המפתחים

■ איך קובעים תקן שיתאים לכל סוגי חברות התוכנה?

☐ האם ניתן לפתח תוכנה איכותית גם מבלי לעמוד בתקן?

מתי צריך תקן?

- ☐ When the software is a product available for the **market** place

- ☐ When the software is used to support the **processes of an organization**

- ☐ When the software is part of a **commercial contract** with another organization

- ☐ When the software is related to **software services**

- ☐ When the software is **embedded in a hardware** product

ISO IEC 90003:2004

- ☐ *[ISO](#) - International Organization for Standardization*.
  - ■ **It was set up in 1947 and is located in Geneva, Switzerland.**
  - ■ **Its purpose is to develop standards that facilitate and support international trade.**

- ☐ *[IEC](#) - International Electrotechnical Commission*.
  - ■ **It was set up in 1906 and is also located in Geneva, Switzerland.**
  - ■ **Its central purpose is to develop standards for all electrotechnologies.**

- ☐ **ISO 9001:2000 quality management standard**

ISO IEC 90003:2004

ISO 9001

+

ADVICE ON HOW TO APPLY ISO 9001 FOR SOFTWARE

= ISO 90003

☐ **ISO 9001:2000 Requirements (*shall* statements)**

☐ **ISO IEC 90003:2004 Guidelines**
  - **Recommendations (*should* statements)**
  - **Suggestions (*may* statements)**

☐ **ההסמכה היא הסמכת ISO 9001**

עיקרי ISO/IEC 90003

- ☐ Systemic Requirements and Guidelines
  - ■ Establish a quality management system for software products
  - ■ Document your software oriented quality system

- ☐ Management Requirements and Guidelines
  - ■ Support quality
  - ■ Focus on your customers
  - ■ Establish a quality policy
  - ■ Perform quality planning
  - ■ Control your quality system
  - ■ Perform management reviews

עיקרי ISO/IEC 90003 (המשך)

- Resource Requirements and Guidelines
  - Provide quality resources
  - Provide quality personnel
  - Provide quality infrastructure
  - Provide quality environment

- Realization Requirements and Guidelines
  - Control software product realization planning
  - Control customer processes
  - Control software design and development
  - Control your purchasing function
  - Manage production and service provision
  - Control monitoring devices

עיקרי ISO/IEC 90003 (המשך)

☐ Remedial Requirements and Guidelines

- ■ Carry out remedial processes
- ■ Monitor and measure quality
- ■ Control your nonconforming software products
- ■ Analyze quality information
- ■ Take required remedial actions

מסמכולוגיה

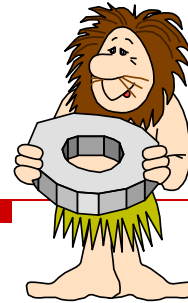☐ אלו הם רק ראשי הפרקים (!). המסמך המלא מכיל את פרוט הדרישות (62 עמודים)

☐ עמידה בתקן מחיבת את ארגון התוכנה לפרט איך הוא עומד בכל אחת מהדרישות

☐ דוגמאות:

■ **ISO IEC 90003 2004 SOFTWARE STANDARD IN PLAIN ENGLISH**

■ **ISO IEC 90003 2004 PLAIN ENGLISH CHECKLIST**

History of 12207

**ISO/IEC 12207 – 1995:   Standard for Information Technology –
Software Life Cycle Processes**

IEEE/EIA 12207.0 – 1996: Software Life Cycle Processes

**IEEE/EIA 12207.1 – 1998: Software Life Cycle Processes –
Life Cycle Data**

**IEEE/EIA 12207.2 – 1998: Software Life Cycle Processes –
Implementation Considerations**

5 PRIMARY
LIFE CYCLE PROCESSES

5.1 Acquisition

5.2 Supply

5.3 Development

5.4 Operation

5.5 Maint.

8 SUPPORTING
LIFE CYCLE PROCESSES

6.1 Documentation

6.2 Configuration Management

6.3 Quality Assurance

6.4 Verification

6.5 Validation

6.6 Joint Review

6.7 Audit

6.8 Problem Resolution

4 ORGANIZATIONAL LIFE CYCLE PROCESSES

7.1 Management

7.2. Infrastructure

7.3 Improvement

7.4. Training

4.1.1.1 Primary Processes [task] (pg 6)

- **Development Process:** **Defines the activities of the developer, the organization that defines and develops the software product.**
  - **Process Implementation**
  - **System Requirements Analysis**
  - **System Architectural Design**
  - **Software Requirements Analysis**
  - **Software Architectural Design**
  - **Software Detailed Design**
  - **Software Coding and Testing**
  - **Software Integration**
  - **Software Qualification Testing**
  - **System Integration**
  - **System Qualification Testing**
  - **Software Installation**
  - **Software Acceptance Support**

5.3.11 System Qualification Testing. This activity consists of the following tasks, which the developer shall perform or support as required by the contract.

5.3.11.1. System qualification testing shall be conducted in accordance with the qualification requirements specified for the system. It shall be ensured that the implementation of each system requirement is tested for compliance and that the system is ready for delivery. The qualification testing results shall be documented.

5.3.11.2. The system shall be evaluated considering the criteria listed below. The results of the evaluations shall be documented.

    a) Test coverage of system requirements.
    b) Conformance to expected results.
    c) Feasibility of operation and maintenance.

5.3.11.3. The developer shall support audit(s) in accordance with 6.7. The results of the audit(s) shall be documented.

5.3.11.4. Upon successful completion of the audit(s), if conducted, the developer shall:

    a) Update and prepare the deliverable software product for Software Installation and Software Acceptance Support.
    b) Establish a baseline for the design and code of each software configuration item.

6.2 Configuration Management Process [a process] (pg. 29):

The Configuration Management Process is a process of applying administrative and technical procedures throughout the software life cycle to: Identify and define software items in a system; control modifications and releases of the items; record and report the status of the items and modification requests; ensure the completeness, consistency, and correctness of the items; and control storage, handling, and delivery of the items.

List of Activities:

6.2.1 Process Implementation [activity]. This activity consists of the following tasks:

6.2.1.1 A configuration management plan shall be developed [task]. The plan shall describe:

- **The CM Activities;**

- **Procedures and Schedule for performing these activities;**

- **The organization(s) responsible for performing these activities;**

- **and their relationship with other organizations, such as software development or maintenance. The plan shall be documented and implemented.**

6.2.2 Configuration Identification [activity]. This activity consists of the following tasks:

6.2.2.1 [task] A scheme shall be established for the identification of software items and their versions to be controlled for the project. For each software CI and its versions, the following shall be identified: the documentation that establishes the baseline; the version references; and other identification details.

6.2.3 Configuration Control. This activity consists of the following task:

6.2.3.1 The following shall be performed: identification and recording of change requests; analysis and evaluation of the changes; approval or disapproval of the request; and implementation, verification, and release of the modified software item. An audit trail shall exist, whereby each modification, the reason for the modification, and authorization of the modification can be traced. Control and audit of all accesses to the controlled software items that handle safety or security critical functions shall be performed.

6.2.4 Configuration Status Accounting. This activity consists of the following tasks:

6.2.4.1 Management records and status reports that show the status and history of controlled software items including baseline shall be prepared. Status reports shall include the number of changes for a project, latest software item versions, release identifiers, the number of releases, and comparisons of releases.

6.2.5 Configuration Evaluation. This activity consists of the following tasks:

6.2.5.1 The following shall be determined and ensured: the functional completeness of the software items against their requirements and the physical completeness of the software items (whether their design and code reflect an up-to-date technical description).
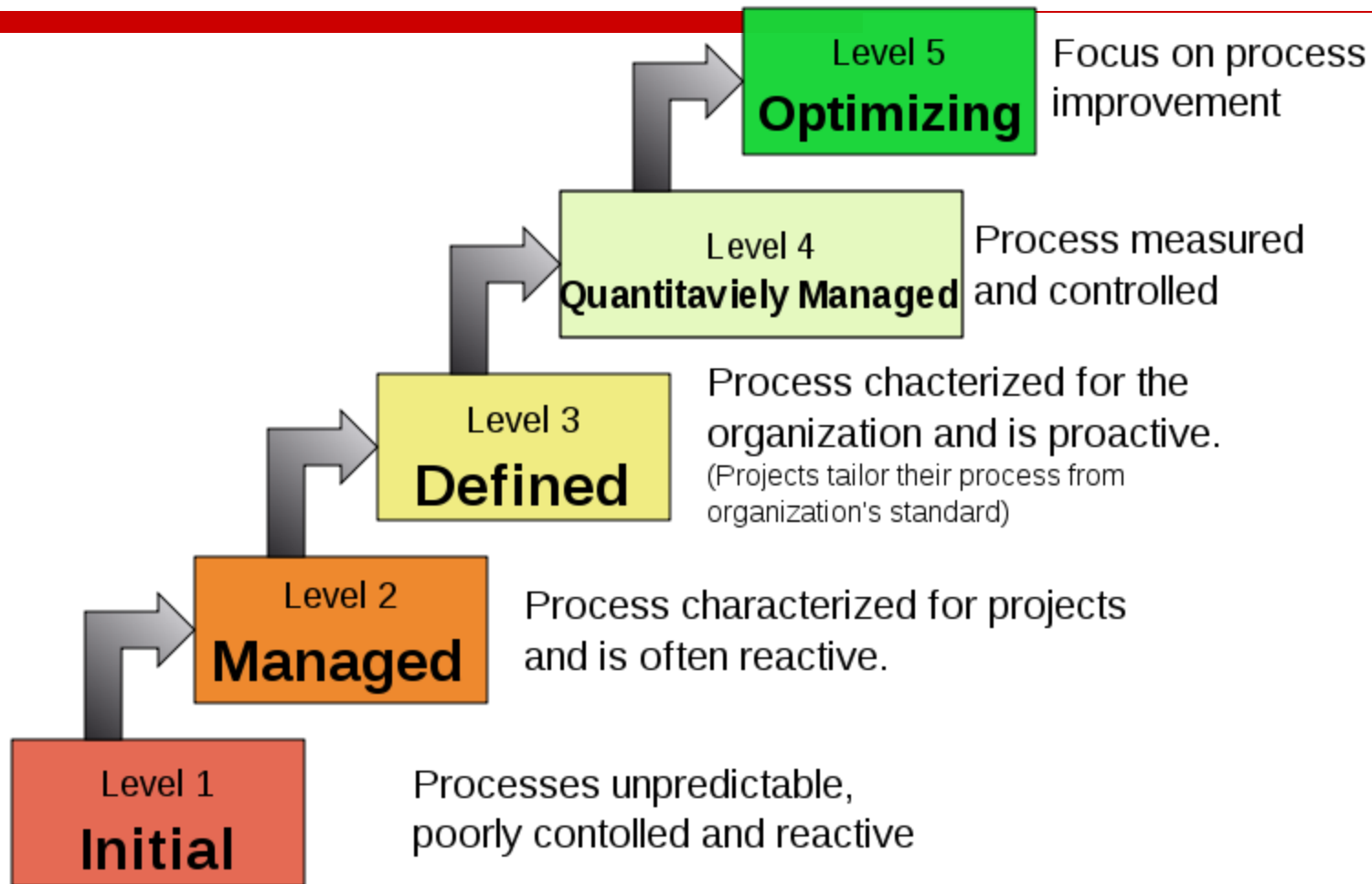
6.2.6 Release Management and Delivery. This activity consists of the following task:

6.2.6.1   The release and delivery of software products and documentation shall be formally controlled. Master copies of code and documentation shall be maintained for the life of the software product. The code and documentation that contain safety or security critical functions shall be handled, stored, packaged, and delivered in accordance with the policies of the organizations involved.

Capability Maturity Model Integration (CMMI)

- A process improvement approach that helps organizations improve their performance. CMMI can be used to guide process improvement across a project, a division, or an entire organization.

- CMMI in software engineering and organizational development is a trademarked process improvement approach

- According to the Software Engineering Institute (SEI, 2008), CMMI helps "integrate traditionally separate organizational functions, set process improvement goals and priorities, provide guidance for quality processes, and provide a point of reference for appraising current processes."

Characteristics of the Maturity levels

Level 5
Optimizing

Focus on process improvement

Level 4
Quantitaviely Managed

Process measured and controlled

Level 3
Defined

Process chacterized for the organization and is proactive.
(Projects tailor their process from organization's standard)

Level 2
Managed

Process characterized for projects and is often reactive.

Level 1
Initial

Processes unpredictable, poorly contolled and reactive

Capability Maturity Model Integrated

Level	Focus	Process Areas	Result
5 Optimizing	Continuous process improvement	Organizational Innovation & Deployment Causal Analysis and Resolution	Productivity & Quality
4 Quantitatively Managed	Quantitative management	Organizational Process Performance Quantitative Project Management	
3 Defined	Process standardization	Requirements Development Technical Solution Product Integration Verification Validation Organizational Process Focus Organizational Process Definition Organizational Training Integrated Project Management Risk Management Decision Analysis and Resolution	
2 Managed	Basic project management	Requirements Management Project Planning Project Monitoring & Control Supplier Agreement Management Measurement and Analysis Process & Product Quality Assurance Configuration Management	
1 Initial	Competent people and heroics		

<http://standards.ieee.org/software/>

730-2002, Standard for Software Quality Assurance Plans

828-1998, Standard for Software Configuration Management Plans

829-1998, Standard for Software Test Documentation

830-1998, Recommended Practice for Software Requirements Specifications

982.1-1988, Standard Dictionary of Measures to Produce Reliable Software

1008-1987 (R1993), Standard for Software Unit Testing

1012-1998, Standard for Software Verification and Validation

1012a-1998, Supplement to Standard for Software Verification and Validation

1016-1998, Recommended Practice for Software Design Descriptions

1028-1997, Standard for Software Reviews

1044-1993, Standard Classification for Software Anomalies

1045-1992, Standard for Software Productivity Metrics

1058-1998, Standard for Software Project Management Plans

1061-1998, Standard for a Software Quality Metrics Methodology

1062-1998, Recommended Practice for Software Acquisition

1063-2001, Standard for Software User Documentation

1074-1997, Standard for Developing Software Life Cycle Processes

1175.1-2002, Guide for CASE Tool Interconnections - Classification and Description

1219-1998, Standard for Software Maintenance

1220-1998, Standard for the Application and Management of the Systems Engineering

1228-1994, Standard for Software Safety Plans

1233-1998, Guide for Developing System Requirements Specifications

1320.1-1998, Standard for Functional Modeling Language-Syntax and Semantics for IDEF0

1320.2-1998, Standard for Conceptual Modeling Language Syntax and Semantics...

1362-1998, Guide for Information Technology-System Definition-Concept of Operations

1420.1-1995, Standard for Information Technology-Software Reuse-Data Model for Reuse

1420.1a-1996, Supplement to Standard for Information Technology-Software Reuse-Data

1420.1b-1999, IEEE Trial-Use Supplement to Standard for Information

1462-1998, Standard - Adoption of International Standard ISO/IEC 14102: 1995; 1465-1998, Standard - Adoption of International Standard ISO/IEC 12119: 1994(E)

1471-2000, Recommended Practice for Architectural Description of Software Intensive Systems

1490-1998, Guide - Adoption of PMI Standard - A Guide to the Project Management Body of Knowledge

פיתוח מערכות תוכנה מבוססות **Java**

בית הספר למדעי המחשב אוניברסיטת תל אביב