April 2012 ניסן התשע"ב – אפריל 2012

ICS CODE: 35.020

# טכנולוגיית מידע - הגדרות מונחים:
# פיתוח מערכות

Information technology - Vocabulary:
System development

**מכון התקנים הישראלי**
**The Standards Institution of Israel**

רח' חיים לבנון 42, תל-אביב 69977, טל' 03-6465154, פקס' 03-6412762, www.sii.org.il 25/06/13 דק

תקן זה הוכן ואושר על ידי הוועדה הטכנית 2110 - מונחי טכנולוגיית המידע, בהרכב זה:

| | | |
|---|---|---|
| אוניברסיטת בר-אילן | - | אריאל פרנק |
| אוניברסיטת תל-אביב | - | אורה קפלן |
| איגוד האינטרנט הישראלי | - | יונתן בן אברהם, דורון שקמוני |
| איגוד ישראלי לעיבוד אינפורמציה (איל"א) | - | מנחם דישון |
| האקדמיה ללשון העברית | - | מלכה זמלי, מלכה מוצ'ניק, אורה שורצולד |
| יבמ ישראל בע"מ | - | שלמה מסטבאום |
| לשכת מנתחי מערכות | - | אמנון שפירא (יו"ר) |
| מיקרוסופט ישראל | - | תמר שגיא |
| רשות השידור | - | דוד גומבוש |
| רשות ההסתדרות לצרכנות | - | יובל פרידמן |

כמו כן תרמו להכנת התקן: אבישי בידני, דורון פינקלשטיין, חיים הירשהורן, יבגני זינגרמן, מיכל שחר, שמשון הופמן, שרה חשבי.

דנית לביא ריכזה את עבודת הכנת התקן.

**הודעה על רוויזיה**

תקן זה בא במקום

התקן הישראלי ת"י 1080 חלק 20 מדצמבר 2003

**מילות מפתח:**

עיבוד נתונים אוטומטי, טכנולוגיית מידע, פיתוח מערכות, מונחים, הגדרות.

**Descriptors:**

automatic data processing, information technology, system development, terms, definitions.

**עדכניות התקן**

התקנים הישראליים עומדים לבדיקה מזמן לזמן, ולפחות אחת לחמש שנים, כדי להתאימם להתפתחות המדע והטכנולוגיה.

המשתמשים בתקנים יוודאו שבידיהם המהדורה המעודכנת של התקן על גיליונות התיקון שלו.

מסמך המתפרסם ברשומות כגיליון תיקון, יכול להיות גיליון תיקון נפרד או תיקון המשולב בתקן.

**תוקף התקן**

תקן ישראלי על עדכוניו נכנס לתוקף החל ממועד פרסומו ברשומות.

יש לבדוק אם התקן רשמי או אם חלקים ממנו רשמיים. תקן רשמי או גיליון תיקון רשמי (במלואם או בחלקם) נכנסים לתוקף 60 יום מפרסום ההודעה ברשומות, אלא אם בהודעה נקבע מועד מאוחר יותר לכניסה לתוקף.

**סימון בתו תקן**

כל המייצר מוצר, המתאים לדרישות התקנים הישראליים החלים עליו, רשאי, לפי היתר ממכון התקנים הישראלי, לסמנו בתו תקן:

**זכויות יוצרים**

© אין לצלם, להעתיק או לפרסם, בכל אמצעי שהוא, תקן זה או קטעים ממנו, ללא רשות מראש ובכתב ממכון התקנים הישראלי.

# תוכן העניינים


הקדמה ..... 1

20.01 - מונחים כלליים - GENRAL TERMS ..... 3

20.02 - חֵקר וניתוח - STUDY AND ANALYSIS ..... 9

20.03 - תֶכֶן מערכות - SYSTEM DESIGN ..... 12

20.04 - מימוש - IMPLEMENTATION ..... 18

20.05 - הבטחת איכות - QUALITY ASSURANCE ..... 20

20.06 - תפעול ותחזוקה - OPERATION AND MAINTENANCE ..... 26

20.07 - תיעוד - DOCUMANTATION ..... 28

20.08 - ניהול מיזמים ובקרתם - PROJECT MANAGEMENT AND CONTROL ..... 31

רשימת מונחים עברית-אנגלית ..... 33

רשימת מונחים אנגלית-עברית ..... 40

**הקדמה**

תקן זה מבוסס על התקן הבין-לאומי ISO/IEC 2382-20 משנת 1999, ומביא מונחים נוספים שנאספו ממקורות שונים. תקן זה הוא חלק מסדרת התקנים הישראליים ת"י 1080, המביאה מונחים והגדרות מהתחומים השונים של טכנולוגיית המידע. סדרת התקנים ת"י 1080 מבוססת על סדרת התקנים הבין-לאומיים ISO 2382 המתעדכנת מפעם לפעם, ועל עבודת איסוף של מונחים חדשים המבוצעת על ידי הוועדה ממקורות שונים. המונחים הנשאבים מהמקורות השונים משמשים להכנת חלקים חדשים או לעדכון חלקים קיימים. המונחים מובאים באנגלית ובעברית, וההגדרות - בעברית.

**חלקי סדרת ת"י** 1080 **שיצאו עד כה לאור הם אלה**:

ת"י 1080 חלק 1 - מונחים בסיסיים
ת"י 1080 חלק 2 - פעולות אריתמטיות ולוגיות
ת"י 1080 חלק 3 - טכנולוגיית חומרה
ת"י 1080 חלק 4 - ארגון נתונים
ת"י 1080 חלק 5 - ייצוג נתונים
ת"י 1080 חלק 6 - הכנת נתונים ועיבודם
ת"י 1080 חלק 7 - תכנות מחשבים
ת"י 1080 חלק 8 - בקרה, כלילות ואבטחה
ת"י 1080 חלק 9 - תקשורת נתונים
ת"י 1080 חלק 10 - טכניקות ואַפשוֹרות של פעולה
ת"י 1080 חלק 11 - בקרה, קלט-פלט וציוד אריתמטי
ת"י 1080 חלק 12 - ציוד היקפי
ת"י 1080 חלק 13 - גרפיקה מחשבית
ת"י 1080 חלק 14 - אמינות, תחזוקתיות וזמינות
ת"י 1080 חלק 15 - שפות תכנות
ת"י 1080 חלק 16 - תורת המידע
ת"י 1080 חלק 17 - מסדי-נתונים
ת"י 1080 חלק 18 - עיבוד נתונים מבוזר
ת"י 1080 חלק 20 - פיתוח מערכות
ת"י 1080 חלק 23 - עיבוד תמלילים
ת"י 1080 חלק 24 - ייצור כליל-מחשב
ת"י 1080 חלק 25 - רשתות מקומיות (LAN)
ת"י 1080 חלק 26 - חיברור מערכות פתוחות (OSI)
ת"י 1080 חלק 27 - המשרד הממוחשב
ת"י 1080 חלק 28 - בינה מלאכותית - מושגי יסוד ומערכות מומחה
ת"י 1080 חלק 29 - בינה מלאכותית - זיהוי דיבור
ת"י 1080 חלק 30 - בינה מלאכותית – ראיית מחשב
ת"י 1080 חלק 31 - בינה מלאכותית - למידת מכונה
ת"י 1080 חלק 32 - דואר אלקטרוני
ת"י 1080 חלק 33 - היפרמדיה ומולטימדיה
ת"י 1080 חלק 34 - בינה מלאכותית – רשתות עצביות
ת"י 1080 חלק 35 - רישות
ת"י 1080 חלק 36 - למידה מתוקשבת
ת"י 1080 חלק 37 - בינה מלאכותית - מציאות מדומה
ת"י 1080 חלק 38 - רשתות חברתיות (בהכנה)
ת"י 1080 חלק 39 - שפות שאילתה (בהכנה)

**זיהוי המונחים**

כל מונח בתקן מסומן בקוד בן 6 ספרות. זוג הספרות הראשון (משמאל) מציין את מספר החלק בסדרה, זוג הספרות השני מציין את מספר הפרק וזוג הספרות השלישי מציין את מספר הסעיף (הפריט).

**הוספות והשמטות בהשוואה לת"י** 20-1080 **משנת** 2003

בתקן זה נוספו מונחים והגדרות שלא נכללו בת"י 20-1080 משנת 2003 (שהיה מבוסס על התקן הבין-לאומי ISO/IEC 2382-20 משנת 1995). התקן מביא מונחים חדשים שנאספו על ידי הוועדה וכן נעשו בו גם כמה תיקוני מונחים בעקבות החלטות האקדמיה ללשון העברית.

**סוגריים מרובעים**

מילה או מילים המופיעות בסוגריים מרובעים, אינן חלק מהמונח. הן מובאות לשם הבהרה בלבד.

**כתיב מלא וכתיב חסר**

המונחים נכתבו בכתיב חסר ובניקוד. ההגדרות נכתבו בכתיב מלא (חסר ניקוד).

**הדגשה**

מילים או ביטויים המודגשים **בעיבוי** בגוף הגדרה הם מונחים.

**רשימות מונחים**

בסוף התקן מובאות שתי רשימות מונחים: רשימת מונחים עברית-אנגלית ורשימת מונחים אנגלית-עברית.

**תחום התקן**

חלק זה של התקן הישראלי ת"י 1080 מביא מונחי פיתוח מערכות והגדרותיהם.

**אזכורים**

**תקנים ישראלים**

ת"י 1080 חלק 1 - טכנולוגית המידע – הגדרות מונחים: מונחים בסיסיים

## 20.01 - מונחים כלליים - Genral Terms

20.01.01 **פִּתּוּחַ מַעֲרָכוֹת** **system development**

תהליך הכולל בדרך כלל ניתוח דרישות, תכן מערכות, מימוש, תיעוד והבטחת איכות.

20.01.02 **תַּהֲלִיךְ פִּתּוּחַ תָּכְנָה** **software development process**
**תָּכְנִית פִּתּוּחַ תָּכְנָה**

תהליך התרגום של צורכי משתמשים למוצר תוכנה. התהליך כולל תרגום צורכי המשתמש לדרישות תוכנה, התמרת דרישות התוכנה לתכן, מימוש התכן בקוד, בדיקת הקוד, ולעתים גם התקנה ובידוק-שחרור של התוכנה לשימוש תפעולי.

**הערה:**

פעילויות אלה יכולות להתבצע במקביל או באופן חִזרורי **(iteratively)**.

20.01.03 **תוכנִית-פרויקט לפרויקט פיתוח תוכנה**. **software development plan**
**SDP** [abb]

תוכנִית-פרויקט לפרויקט פיתוח תוכנה.

20.01.04 **מַחֲזוֹר פִּתּוּחַ תָּכְנָה** **software development cycle**

התקופה הפותחת בהחלטה לפתח מוצר תוכנה והמסתיימת במסירתו לשימוש. מחזור זה כולל בדרך כלל שלב דרישות, שלב תכן, שלב מימוש, שלב בדיקה, ולעתים גם שלב התקנה ושלב בידוק-שחרור.

**הערות:**

1. שלבים אלה יכולים להתבצע במקביל או באופן חזרורי, לפי גישת הפיתוח המשמשת.
2. מונח זה משמש לעתים לתיאור תקופה ארוכה יותר, המתחילה בניתוח הצורך לפתח מוצר והמסתיימת כאשר מפסיקים להרחיב ולשפר את התוכנה או כאשר מסתיים מחזור החיים השלם שלה.

20.01.05 **דְּרִישָׁה** **requirement**

תנאי שיש למלא, או יכולת שיש להקנות למערכת או לרכיב של מערכת, כדי לעמוד בחוזה, בתקן, במפרט או במסמך רשמי מחייב אחר.

20.01.06 **מִפְרָט** **specification**

מסמך המפרט בשלמות, במדויק ובאופן הניתן לאימות, את הדרישות, התכן, ההתנהגות או אופיינים אחרים של מערכת או של רכיב, ולעתים קרובות גם את הנהלים לקביעה אם תנאים אלה מולאו.

20.01.07 **מִפְרָט פּוֹרְמָלִי** **formal specification**

מִפרָט המשמש כדי להוכיח מתמטית את תקפות המימוש, או כדי להסיק מתמטית את נכונות המימוש.

20.01.08 **מִפְרָט מוּצָר** **product specification**

מסמך המפרט את התכן של מערכת או של רכיב, החייב להתממש במוצר המוגמר.

20.01.09 **קַו בָּסִיס** **baseline**

בפיתוח תוכנה - הסכם או תוצאה שהוגדרו ונקבעו בזמן נתון, שממנו ואילך כל שינוי דורש הצדקה ואישור.

20.01.10 **יִשּׁוּם** **application**

עבודה מוגדרת, שמשתמש מבצע במחשב.

**דוגמות:**

יישום חישוב שכר, מערכת הזמנת מקומות לטיסה.

20.01.11 **בְּעָיַת יִשּׁוּם** **application problem**

מטלה שמשתמש מגיש ליישום והדורשת עיבוד מידע לשם פתרונה.

20.01.12 **תָּכְנַת יִשּׁוּמִים** **application software**

תוכנה שנועדה לשרת יישום אחד או יותר.

**דוגמות**:

תוכנת תמיכה, תוכנת מערכת.

20.01.13 **תָּכְנִית יִשּׁוּם** **application software**
**application program**

תוכנית שנועדה לפתור בעיית יישום.

**דוגמות**:

תוכנית חישוב שכר, תוכנית ניהול חשבונות.

20.01.14 **מוּצַר תָּכְנָה** **software product**

אוסף של תוכניות מחשב, נהלים, תיעוד מסומך ולעתים גם נתונים מסומכים, המיועדים למסירה למשתמש.

20.01.15 **תָּכְנַת תְּמִיכָה** **support software**

תוכנה שנתכנה כדי לאפשר פיתוח ותִחזוק של תוכנה אחרת.

**דוגמות**:

מהדרים, תוכניות עריכה, תוכניות טעינה ותוכניות שירות אחרות.

20.01.16 **מַעְלָה-מַטָּה** **top-down**

מתייחס לשיטה או לנוהל המתחילים ברמת ההפשטה הגבוהה ביותר ויורדים עד לרמת ההפשטה הנמוכה ביותר.

20.01.17 **מַטָּה-מַעְלָה** **bottom-up**

מתייחס לשיטה או לנוהל המתחילים ברמת ההפשטה הנמוכה ביותר ועולים עד לרמת ההפשטה הגבוהה ביותר.

20.01.18 **מַעֲרֶכֶת סְגוּרָה** **closed system**

מערכת שאין בה לפחות אחד האופיינים האלה: יבילות **(portability)**, תפעוליות בֵּינית **(interoperability)** וקישוריות.

20.01.19 **שְׁלַב פִּתּוּחַ מַעֲרֶכֶת** **system development phase**

השלב שתחילתו בהחלטה לפתח מערכת וסופו מסירת המערכת לַמשתמש.

20.01.20 **שֶׁלֶב בְּמַחֲזוֹר חַיִּים** **lifecycle phase**

כל אחת מהפעילויות המרכיבות את מחזור החיים של מערכת

**דוגמות**:

שלב ההמשגה, שלב המימוש, שלב ההתקנה ובידוק השחרור, שלב הייצור, שלב הדרישות, שלב הפרישה, מחזור פיתוח התוכנה, פיתוח המערכת, שלב הבדיקה.

20.01.21 **מַחֲזוֹר חַיֵּי מַעֲרֶכֶת** **system lifecycle**

מהלך התפתחות מערכת מראשיתה עד זניחתה

**הערה**:

במחזור חיי מערכת ניתן לזהות שלבים כגון ניתוח דרישות, תכן, מימוש, בדיקה, התקנה ובידוק שחרור, תפעול ותחזוקה ולעתים גם גמר שימוש.

20.01.22 **שְׁלַב הַדְּרִישׁוֹת** **requirements phase**

החלק במחזור החיים, שמגדירים ומתעדים בו את הדרישות מהמוצר.

20.01.23 **שְׁלַב הַהַמְשָׁגָה** **concept phase**

החלק במחזור החיים, שמפתחים בו את הרעיונות והמושגים המופשטים המתייחסים להיבטים הכלליים של המוצר, לרבות הֶקשר, מִנשקים, תזרים הפעילויות בעת השימוש והארכיטקטורה.

20.01.24 **שְׁלַב הַתְּכָן** **design phase**

החלק במחזור החיים שמפיקים בו את התכן של הארכיטקטורה, של היחידות התפקודיות, של המִנשקים ושל הנתונים, מתעדים את התכן ומאמתים את עמידתו בדרישות.

20.01.25 **תְּכָן רִאשׁוֹנִי** **preliminary design**

תהליך הניתוח של חלופות התכן והגדרת הארכיטקטורה, הרכיבים, המִנשקים ואומדני הזמן והגודל של המערכת או של רכיביה.

20.01.26 **מְצַאי נְתוּנִים** **data inventory**

במערכת עיבוד מידע - כל הנתונים והאופיינים שלהם, לרבות יחסי התלות ההדדית.

20.01.27 **תְּכָן מְפוֹרָט** **detailed design**

תהליך הפירוט וההרחבה של התכן הראשוני של מערכת או של רכיב, המספיק למימושם או התוצאה של תהליך זה.

20.01.28 **אַב-טִיפּוּס** **prototype**

מודל או מימוש ראשון, המשמשים להערכה של תכן מערכת, ביצועיה וישימותה, או להבנה או לקביעה טובות יותר של הדרישות.

**הערה**:

רבים: אב-טיפוסים.

20.01.29 **הַטְפָּסַת-אָב** **prototyping**

שיטה לפיתוח חומרה ותוכנה, שבה מפתחים גרסה ראשונית, מלאה או חלקית, של חומרה או של תוכנה, כדי לאפשר משוב של משתמשים או כדי לחקור זמנים והיתכנות או נושאים אחרים, לתמיכה בתהליך הפיתוח.

20.01.30 **פְּרוֹיֶקְט חָלוּץ** **pilot project**

פרויקט שמטרתו לבחון גִרסה ראשונה של מערכת עיבוד מידע בתנאי פעולה ממשיים אך מוגבלים. גִרסה זו משמשת אחר כך לבדיקת ההגדרה המוגמרת של המערכת.

20.01.31 **שְׁלַב הַבְּדִיקָה** **test phase**

התקופה במחזור חיי התוכנה שבה מעריכים את רכיבי מוצר התוכנה, משלבים אותם ומעריכים את מוצר התוכנה כולו, כדי לקבוע אם הוא ממלא אחר הדרישות.

20.01.32 **הַדְמָיָה** **Simulation**

שימוש במערכת עיבוד נתונים כדי לייצג אופייני התנהגות נבחרים של מערכת פיזית או מופשטת.

**דוגמה**:

הייצוג של זרימת אוויר סביב עליינים אווירודינמיים (airfoils) במהירויות, בטמפרטורות ובלחצי אוויר שונים.

20.01.33 **שְׁלַב (הַ)הַתְקָנָה וּבְדוּק (הַ)שִׁחְרוּר** **installation and checkout phase**

התקופה במחזור חיי התוכנה, שבה משלבים מוצר תוכנה בסביבתו התפעולית ובודקים אותו בסביבה זו, כדי להבטיח שהוא מתפקד כנדרש.

20.01.34 **שְׁלַב הַמִּמּוּשׁ** **implementation phase**

החלק (החלקים) במחזור חיים, (שבו/שבהם) יוצרים או רוכשים אותה ובודקים מערכת או יחידה תפקודית.

20.01.35 **מִמּוּשׁ מַעֲרֶכֶת** **implementation of a system**

**מִמּוּשׁ** **implementation**

השלב בפיתוח מערכת, שבסופו החומרה, התוכנה ונוהלי המערכת הופכים תפעוליים.

20.01.36 **שְׁלַב הַתִּפְעוּל וְהַתַּחְזוּקָה** **operation and maintenance phase**

התקופה במחזור חיי התוכנה, שבה מוצר התוכנה מתפקד בסביבתו התפעולית, מנוטר לביצועים משביעי רצון ומוסגל (modifiied) כנדרש לתיקון בעיות או כמענה לשינויים בדרישות.

20.01.37 **גִּרְסָה** **version**

בפיתוח מערכות - מהדורה ראשונה או מחודשת של פריט תצורת תוכנה, המלֻוָה בהידור מלא או בהידור-מחדש מלא של פריטי תצורת התוכנה.

20.01.38 **תְּמִיכָה בַּמַּעֲרֶכֶת** **system support**

הספקה שוטפת של שירותים וחומרים הנחוצים לשימוש במערכת ממומשת ולשיפורה.

20.01.39 **שְׁלַב הַזְּנִיחָה** **retirement phase**

התקופה במחזור חיי המערכת שבה התמיכה במוצר מופסקת.

20.01.40 **סְטִיָה** **deviation**

חריגה מדרישה.

**הערות**:

1. לעתים משתמשים במונח ככינוי ל.
   הרשאה בכתב לחרוג מדרישת ביצועים או מדרישת תכן מיוחדות, לגבי מספר יחידות מוגדר או למשך זמן מוגדר.
   ההרשאה מוענקת לפני תהליך הפיתוח או הייצור של פריט, או במהלכם.
2. שלא כמו שינוי הנדסי, סטייה אינה דורשת תיקון של התיעוד המגדיר את הפריט הנידון.

20.01.41 **נְצִילוּת** **efficiency**

היחס שבין השימוש במשאב לבין השימוש הכולל שלו.

**דוגמה**:

נצילות האחסון של מסד נתונים היא היחס שבין נתוני המשתמש לבין סך כל הנתונים, כאשר בסך כל הנתונים נכללים נתוני תקורה כגון מצביעים ומפתחות.

20.01.42 **נְצִילוּת בְּצוּעַ** **execution efficiency**

מדד כמותי כלשהו (מטריקה), שהתקבל באמצעות התייחסות לביצועי תוכנית.

20.01.43 **כּוֹשֶׁר הִתְרַחֲבוּת** **extendability**

**מִתְרַחֲבוּת** **extensibility**

הקלות שבה ניתן להסגיל (to modify) יחידה תפקודית כדי להגדיל את קיבולת האחסון שלה או את כושר התפקוד שלה.

20.01.44 **תַּחְזוּקָתִיוּת** **maintainability**

הקלות שבה ניתן להסגיל (to modify) יחידה תפקודית כדי לתקן תקלות, כדי לשפר ביצועים או תכונות אחרות, כדי להתאים אותה לסביבה שהשתנתה או כדי להוסיף לה תפקודים.

20.01.45 **רַב-שִׁמּוּשִׁיּוּת** **reusability**

**generality**

המידה שבה ניתן להשתמש במודול תוכנה או בתוצר תוכנה אחר ביותר מתוכנית מחשב אחת או ממערכת תוכנה אחת.

20.01.46 **רַב-שִׁמּוּשִׁי** **reusable**

מתייחס למודול תוכנה או לתוצר תוכנה אחר, שניתן להשתמש בהם ביותר מתוכנית מחשב אחת או ממערכת תוכנה אחת.

20.01.47 **חֹסֶן** **robustness**

המידה שבה מערכת או רכיב יכולים לתפקד נכונה בנוכחות קלטים שגויים או בתנאי לחץ סביבתי.

20.01.48 **מְתֹאָר מֵאֵלָיו** **self-descriptive**

מתייחס למערכת או לרכיב, המכילים די מידע כדי להסביר את המטרות ואת המאפיינים שלהם.

20.01.49 **מְתֹעָד מֵאֵלָיו** **self-documented**

מתייחס לקוד-מקור המכיל הערות המסבירות את מטרותיו ותפעולו, וכל מידע מועיל אחר המשמש להבנת הקוד ולתִחזוקו.

20.01.50 **שֵׁרוּתִיּוּת** **serviceability**

הקלות שבה ניתן לקיים יחידה תפקודית במצב שבו תוכל לתפקד כנדרש ממנה, או הקלות שבה ניתן להחזיר אותה למצב כזה.

20.01.51 **שִׂדּוּד** **sizing**

תהליך הערכת קיבולת האחסון של המחשב או ומספר שורות המקור, הדרושים למערכת תוכנה או לרכיב תוכנה.

20.01.52 **בָּדִיקָתִיּוּת** **testability**

המידה שבה מערכת או רכיב, או דרישות הפיתוח שלהם, מאפשרים לקבוע קריטריוני בדיקה ולערוך בדיקות כדי לקבוע אם הקריטריונים מולאו.

20.01.53 **תִּזְמוּן** **timing**

תהליך ה אומדן או המדידה של כמות זמן הביצוע הדרושה למערכת תוכנה או לרכיב תוכנה.

20.01.54 **נַתַּח תִּזְמוּן** **timing analyzer**

כלי תוכנה האומד או המודד את זמן הביצוע של תוכנית מחשב או של חלק מתוכנית מחשב, אם על ידי סיכום זמני הביצוע של ההוראות לאורך נתיבים מסוימים ואם על ידי תחיבה של גשושות (probes) בנקודות מוגדרות בתוכנית ומדידת זמן הביצוע שבין הגשושות.

20.01.55 **שִׁמּוּשִׁיּוּת** **usability**

הקלות שבה משתמש יכול ללמוד כיצד לתפעל מערכת או רכיב, להזין קלט ולפרש את הפלט.

20.01.56 **מִנְשָׁק מִשְׁתַּמֵּשׁ** **user interface**

**human-machine interface**

**HMI** [abb]

**man-machine interface**

**MMI** [abb]

**human-computer interface**

**HCI** [abb]

מִנשק המאפשר העברת מידע בין משתמש אנושי לבין רכיבי החומרה או התוכנה של מערכת מחשב.

20.01.57 **יְבִילוּת מִשְׁתַּמֵּשׁ** **user portability**

הקלות שבה אנשים יכולים לעבוד במקומות שונים או במערכות שונות ללא צורך באימון מחדש.

20.01.58 **נִיצוּל** **utilization**

היחס שבין השימוש במשאב לזמינות שלו.

**דוגמה**:

בהערכת ביצועי מחשב – היחס בין פרק הזמן שבו המערכת או הרכיב "תפוסים" לבין פרק הזמן שבו הם זמינים.

20.01.59 **תָּכְנַת מַעֲרֶכֶת** **system software**

תוכנה שאינה תלויה ביישום, המסייעת להרצת יישומי תוכנה.

**דוגמות**: מערכות הפעלה, מאספים, תוכניות שירות.

20.01.60 **תָּכְנָה קְרִיטִית** **critical software**

תוכנה שכֶּשֶל שלה עלול להשפיע על הבטיחות או לגרום הפסד פיננסי או חברתי ניכר.

20.01.61 **חֲבִילַת תָּכְנָה** **software package**

אוסף שלם ומתועד של תוכניות המסופקות לכמה משתמשים, ליישום או לתפקוד בעלי אופי דומה.

20.01.62 **מַחֲזוֹר חַיֵּי תָּכְנָה** **software lifecycle**

מסלול שינויי הפיתוח שתוכנה עוברת למן העלאת הרעיון ועד סיום השימוש.

20.01.63 **סֵפֶר** **reference book**

**eference guide**

**reference manual**

פרסום המשמש לעיון ולרכישת ידע על התורה והתפקוד של מערכת עיבוד מידע.

**הערה**:

בדרך כלל יאורגן הספר אלף-ביתי ויכלול מידע תמציתי בכל נושא.

**דוגמות**:

ספר פקודות של שפת מחשב, תיאור תפקודי של מערכת מלאי, תיאור רכיבי חומרה/תוכנה, ספר הודעות השגיאה וכדומה.

20.01.64 **שִׁמּוּשׁ חוֹזֵר** **reuse**

השימוש בתכן מערכת או בקוד תוכנית, בחלקם הגדול או במלואם, ביישום אחר.

20.01.65 **תֶּכֶן כִּוּוּן-עֲצָמִים** **object-oriented design**

שיטת תכן תוכנה הממדלת את האופיינים של עצמים מופשטים או ממשיים באמצעות מחלקות ועצמים.

20.01.66 **הוֹרֶה** **parent**

ישות ברמה מעל ישות אחרת במדרג שיש בינן יוחסה.

20.01.67 **צֶאֱצָא** **descendant**

ישות ברמה מתחת לישות אחרת במדרג שיש בינן יוחסה.

20.01.68 **מָסַר** **deliver** (v)

הגיש את תוצאות עבודתו עד אבן-דרך מוסכמת.

20.01.69 **נִמְסָר** **deliverable**

תוצאות עבודות הפיתוח המוגשות כאשר מגיעים לאבן-דרך מוסכמת.

## 20.02 - חֵקר וניתוח - Study and Analysis

20.02.01 **חֵקֶר הִזְדַּמְּנוּת** **opportunity study**

חקר שמטרתו לבחון בעיה ולקבוע אם יש לפותרה בפרק הזמן הנידון.

20.02.02 **חֵקֶר הִתְּכְנוּת** **feasibility study**

חקר שמטרתו לזהות ולנתח בעיה ואת פתרונותיה האפשריים, כדי לקבוע אם ניתן ליישם אותם וכדי לקבוע את עלותם ואת תועלתם.

**הערה**:

המונח "חקר ישימות" אינו מומלץ.

20.02.03 **הַגְדָּרַת בְּעָיָה** **problem definition**

**תֵּאוּר בְּעָיָה** **problem description**

הצגת בעיה, העשויה לכלול תיאור הנתונים, ואת השיטה, הנהלים, והאלגוריתמים המשמשים לפתרונה.

20.02.04 **הֶפְשֵׁט [ש"ע]** **abstraction (1)**

נקודת ראות לגבי עצם, המתמקדת במידע המתייחס למטרה מסוימת ומתעלמת משאר המידע.

20.02.05 **הַפְשָׁטָה [שם פעולה]** **abstraction (2)**

תהליך הניסוח של **הֶפְשֵׁט**.

20.02.06 **נִתּוּחַ דְּרִישׁוֹת** **requirements analysis**

חקירה שיטתית של דרישות המשתמש כדי להגיע להגדרה של מערכת.

20.02.07 **דְּרִישָׁה סְבִיבָתִית** **environmental requirement**

דרישה המפרטת את התנאים או את האילוצים הסביבתיים שבתוכם חייבות לפעול המערכת או היחידה התפקודית.

**דוגמות**:

טמפרטורה, לחות, מרחב, עומס רצפה, רטט.

20.02.08 **דְּרִישָׁה תִּפְקוּדִית** **functional requirement**

דרישה המפרטת פונקציות שמערכת או רכיב חייבים לבצע.

20.02.09 **דְּרִישַׁת מִמּוּשׁ** **implementation requirement**

דרישה המפרטת את תהליך היצירה, הרכישה והבדיקה של מערכת או של יחידה תפקודית או המטילה עליו אילוצים.

**דוגמה**: שימוש בשפת תִכנות מסוימת, שימוש בנוהלי הבטחת איכות מיוחדים.

20.02.10 **דְּרִישַׁת מִנְשָׂק** **interface requirement**

דרישה המפרטת פריט חיצוני שאיתו חייבים המערכת או הרכיב במערכת להדד, או המטילה אילוצים על תסדירים, על תזמון או על גורמים אחרים הנובעים מהידוד כזה.

20.02.11 **דְּרִישַׁת בִּצּוּעִים** **performance requirement**

דרישה האוכפת תנאים על ביצועי יחידה תפקודית.

**דוגמות**:

מהירות, דיוק, ניצול זיכרון.

20.02.12 **דְּרִישָׁה פִּיזִיקָלִית** **physical requirement**

דרישה המפרטת אופיין פיזיקלי של מערכת או של רכיב מערכת.

**דוגמות**:

חומר, צורה, גודל, משקל.

20.02.13 **תַּסְקִיר דְּרִישׁוֹת** **requirements review**

תהליך או פגישה שבמהלכם מוצגות הדרישות עבור מערכת, פריט חומרה או פריט תוכנה לפני צוות הפרויקט, מנהלים, משתמשים, לקוחות או צדדים מעוניינים אחרים, למתן הערות או לאישור.

**דוגמה**:

תסקיר דרישות מערכת, תסקיר דרישות תוכנה.

**הערה**:

שם הפעולה - סקירה.

20.02.14 **תַּסְקִיר דְּרִישׁוֹת תָּכְנָה** **software requirements review**

**תַּסְקִיר מִפְרַט תָּכְנָה** **SRR** [abb]

**software specification review**

**SSR** [abb]

תסקיר הדרישות שהוגדרו עבור פריט תצורת תוכנה אחד או יותר, כדי להעריך עד כמה הם עומדים בדרישות המערכת וכיצד הם מפרשים אותן, וכן כדי לקבוע אם יש בהם בסיס מספק כדי להמשיך בתכן שלהם.

20.02.15 **נִתּוּחַ מַעֲרָכוֹת** **system analysis**

**systems analysis**

חקירה שיטתית של מערכות קיימות או מתוכננות, כדי לקבוע את דרישות המידע ואת תהליכי המערכות וכיצד הם מתייחסים זה לזה ולמערכות אחרות.

20.02.16 **נִתּוּחַ תִּפְקוּדִי** **functional analysis**

חקירה שיטתית של תפקודי מערכת קיימת או מתוכננת.

20.02.17 **נִתּוּחַ מֵידָע** **information analysis**

חקירה שיטתית של מידע וזרימתו במערכת קיימת או מתוכננת.

20.02.18 **נִתּוּחַ נְתוּנִים** **data analysis**

חקירה שיטתית של נתונים וזרימתם במערכת קיימת או מתוכננת.

20.02.19 **נִתּוּחַ נְתִיבִים** **path analysis**

ניתוח של תוכנית מחשב כדי לזהות מעקובות ביצוע אפשריות דרך התוכנית, שכיחות השימוש בהן, אורכיהן או תכונות אחרות שלהן.

20.02.20 **תְּנָאֵי נָתִיב** **path condition**

קבוצת תנאים שחייבים להתקיים כדי ליזום מעקובת ביצוע.

20.02.21 **בִּטוּי נָתִיב** **path expression**

ביטוי לוגי המגדיר תנאי נתיב.

20.02.22 **תַּרְשִׁים מְמֻרְכָּז מִבְנֵה-נְתוּנִים** **data-structure-centered diagram**

טכניקת תכן תוכנה, שבה ארכיטקטורת המערכת נגזרת מניתוח המבנה של קבוצות הנתונים שהמערכת חייבת לעסוק בהן.

20.02.23 **תַּרְשִׁים מִבְנֵה נְתוּנִים** **data-structure diagram**

תרשים המתאר קבוצת אלמנטי נתונים, מאפייניהם והיחסות הלוגיות ביניהם.

20.02.24 **תַּרְשִׁים יְשׁוּיוֹת-יַחְסוֹת** **entity-relationship diagram, E-R diagram**
**מַפַּת יְשׁוּיוֹת-יַחְסוֹת** **entity-relationship map, E-R map**

תרשים המתאר קבוצת ישויות ממשיות והיוחסות הלוגיות שביניהן.

20.02.25 **רֶשֶׁת פֶּטְרִי** **petri net**

מודל פורמלי מופשט של זרימת מידע, המציג תכונות סטטיות ודינמיות של מערכת, והמוצג בדרך כלל בתרשים, הכולל:

א - שני טיפוסי צמתים (המכונים "מקומות" ו"מעברים") המחוברים באמצעות קשתות;

ב - סימנים (המכונים "אסימונים") המציינים תכונות דינמיות.

20.02.26 **מַעֲקַב מַעֲרֶכֶת** **system follow-up**
**תַּסְקִיר אַחַר מִמּוּשׁ** **post-implementation review**
**post-development review**

חקר התוצאים **(effects)** של מערכת לאחר שהתייצבה מבחינת השימוש התפעולי.

20.02.27 **מוֹדֵל יֵשֵׁיוֹת-יַחְסוֹת** **entity-relationship model**

מודל נתונים המתאר קבוצת ישויות ואת היוחסות ביניהן באמצעות **תרשימי ישויות-יוחסות**.

20.02.28 **מוֹדֵל מֵידַע פָּתוּחַ** **Open Information Model**
**OIM**

מודל עצמים עבור מידע מסוגים מיוחדים הגמיש דיו לתמיכה בסוגי מידע חדשים.

20.02.29 **אַמְדָן** **Estimator**

כלי המשתמש בנתונים סטטיסטיים כדי להגיע לאומדן הברירות, העוצמה והעלות של מתווי ביצוע.

## 20.03 - תֶּכֶן מערכות - System Design

20.03.01 **תֶּכֶן מַעֲרָכוֹת** **system design**

תהליך ההגדרה של ארכיטקטורת חומרה ותוכנה, של רכיבים, של מודולים, של מִנשקים ושל נתונים של מערכת, הדרושים כדי לספק דרישות מוגדרות.

20.03.02 **תֶּכֶן אַרְכִיטֶקְטוֹנִי** [הפעולה] **architectural design**

תהליך ההגדרה של אוסף רכיבי חומרה ותוכנה והמִנשקים שביניהם, כדי לקבוע את המסגרת לפיתוח מערכת מחשב.

20.03.03 **תֶּכֶן אַרְכִיטֶקְטוֹנִי** [התוצאה] **architectural design**

התוצאה של תהליך ההגדרה של אוסף רכיבי חומרה ותוכנה והמִנשקים שביניהם, הקובע את המסגרת לפיתוח מערכת מחשב.

20.03.04 **תֶּכֶן מַעֲרָכוֹת מֻשְׁגִי** **conceptual system design**

רכיב בתֶּכֶן מערכת, העוסק בהיבטים הלוגיים של ארגון המערכת, בתהליכים שבה ובזרימת המידע בתוכה.

20.03.05 **תֶּכֶן תִּפְקוּדִי** **functional design**

הפונקציות של רכיבי המערכת ושל יוחסות העבודה ביניהן.

20.03.06 **תֶּכֶן מֻבְנֶה** **structured design**

כל גישה שיטתית לתכן תוכנה, הדבֵקה בכללים מוגדרים המבוססים על עקרונות כגון מודולריות, תכן מעלה-מטה ועידון צעד-צעד של נתונים, מבני מערכת וצעדי עיבוד, וכמו כן תוצאותיה של גישה זו.

20.03.07 **תֶּכֶן כִּוּוּן עֲצָמִים** **object-oriented design**

טכניקת פיתוח תוכנה, שבה מבטאים מערכת או רכיב על ידי עצמים והחיבורים ביניהם.

20.03.08 **תֵּאוּר תֶּכֶן תָּכְנָה** **software design description**

**SDD** [abb]

ייצוג של תוכנה הנוצר כדי לאפשר ניתוח, תכנון, מימוש וקבלת החלטות, והמשמש סרטוט או דגם של המערכת וכלי להעברת מידע על תיאור התכן של התוכנה.

20.03.09 **יְחִידַת תֶּכֶן** **design unit**

אוסף קשור לוגית של אלמנטי תכן.

20.03.10 **פְּרִיט תְּצוּרָה** **configuration item**

**CI** [abb]

צבר יחידות תפקודיות או כל חלק בדיד שלהן, המטופל כישות יחידה בתהליך ניהול תצורה.

20.03.11 **פְּרִיט תְּצוּרַת תָּכְנָה** **computer software configuration item**

**CSCI** [abb]

**computer program configuration item**

**CPCI** [abb]

תוכנה המטופלת כישות יחידה בתהליך ניהול תצורה.

20.03.12 **פְּרִיט תְּצוּרַת חֹמְרָה** **hardware configuration item**

**HWCI** [abb]

צבר חומרה המטופל כישות יחידה בתהליך ניהול תצורה.

20.03.13 **פְּרִיט קְרִיטִי** **critical item**

בניהול תצורה - פריט בתצורה, שעקב שיקולים הנדסיים או לוגיסטיים מיוחדים דורש מפרט מאושר לכינון בקרה טכנית או בקרת מלאי ברמת הרכיב.

20.03.14 **הַפְשָׁטַת נְתוּנִים** **data abstraction**

תהליך המיצוי של אופייני-הנתונים העיקריים, על ידי הגדרה של טיפוסי הנתונים ושל האופיינים התפקודיים שלהם והתעלמות מפרטי ייצוג.

**הערה**:

המונח מתייחס גם לתוצאת התהליך המתואר לעיל.

20.03.15 **חֵלֶק קְרִיטִי תְּחִילָה** **critical piece first**

גישת פיתוח מערכות שבה ההיבטים הקריטיים ביותר של מערכת ממומשים תחילה.

**הערה**:

ניתן להגדיר את החלק הקריטי במונחים של השירותים המסופקים, רמת הסיכון, הקושי או קריטריונים אחרים.

20.03.16 **פִּתּוּחַ תּוֹסַפְתִּי** **incremental development**

טכניקת פיתוח תוכנה שבה הגדרת הדרישות, התכן, המימוש והבדיקה מתרחשים בחפייה ובאופן חִזרורי **(iterative),** כך שהתוצאה שלהם היא השלמה תוספתית של מוצר התוכנה המלא.

20.03.17 **עִדּוּן צַעַד-צַעַד** **stepwise refinement**

טכניקת פיתוח תוכנה שבה מגדירים את הנתונים ואת צעדי העיבוד בהגדרה כללית תחילה, ולאחר מכן – ביתר פירוט.

20.03.18 **הַטְפָּסַת-אָב מְהִירָה** **rapid prototyping**

סוג הטפסת-אב שבו מתמקדים בפיתוח אב-טיפוסים בשלב מוקדם בתהליך הפיתוח, כדי לקבל משוב וניתוח מוקדמים כתמיכה בתהליך הפיתוח.

20.03.19 **קֶלֶט-עִבּוּד-פֶּלֶט** **input-process-output**

**IPO** [abb]

טכניקת תכן תוכנה המורכבת מזיהוי הצעדים הכרוכים בכל תהליך שיש לבצע ומזיהוי הקלט והפלט של כל צעד.

**הערה**:

עידון של שיטה זו, המכונה "קלט-עיבוד-פלט מדרגי", מזהה את הצעדים, הקלט והפלט ברמה הכללית וגם ברמות הפירוט.

20.03.20 **צַנֶּרֶת** **Pipeline**

טכניקה של תכן תוכנה או של תכן חומרה, שבה הפלט של תהליך אחד משמש כקלט לתהליך שני וכך הלאה, לעתים קרובות בו-זמנית בתוך מחזור זמן יחיד.

**דוגמות**:

1. בפיתוח מערכות מאפשרת טכניקה זו ביצוע סדרתי או בו-זמני של שלבים בתכן של חומרה או של תוכנה;
2. ביחידות עיבוד מרכזיות של מחשב **(CPU)** מאפשרת טכניקה זו ביצוע בו-זמני של חלקי פקודת מחשב**,** כגון: ביצוע שלב הקריאה של הפקודה הבאה לפני שהפקודה הנוכחית הסתיימה;
3. במתגי תקשורת מאפשרת טכניקה זו ביצוע סדרתי או בו-זמני של תהליכי קלט, ניתוב ופלט.

20.03.21 **מוֹדֵל סְפִּירָלִי** **spiral model**

מודל של תהליך פיתוח תוכנה שבו הפעילויות המרכיבות, כגון ניתוח דרישות, תכן ראשוני ותכן מפורט, קידוד, כילול ובדיקה, מבוצעות באופן חִזרורי עד השלמת הפיתוח.

20.03.22 **מוֹדֶל מַפַּל-מַיִם** **waterfall model**

מודל של תהליך פיתוח תוכנה שבו הפעילויות המרכיבות, כגון שלב ההמשגה, שלב הדרישות, שלב התכן, שלב המימוש, שלב הבדיקה ושלב ההתקנה ובידוק השחרור, מבוצעות בסדר המפורט, ייתכן שבחפייה אך ללא חִזרור או במעט חִזרור.

20.03.23 **תְּנוּעָה** **transaction**

בפיתוח מערכות – אלמנט נתונים, אלמנט בקרה, אות, אירוע או שינוי מצב, הגורמים, המזניקים או המְתַחלים פעולה או מעקובת של פעולות.

20.03.24 **נִתּוּחַ תְּנוּעוֹת** **transaction analysis**

**תֶּכֶן מְמֻרְכַּז-תְּנוּעוֹת** **transaction-centered design**

טכניקת פיתוח תוכנה שבה מבנה המערכת נגזר מניתוח התנועות שהמערכת נדרשת לעבד.

20.03.25 **נִתּוּחַ הַתְּמָרָה** **transform analysis**

**תֶּכֶן מְמֻרְכַּז-הַתְּמָרָה** **transformation analysis**

**transform-centered data**

טכניקת פיתוח תוכנה שבה מבנה המערכת נגזר מניתוח זרימת הנתונים דרך המערכת וההתמרות שהנתונים חייבים לעבור.

20.03.26 **אִחוּד מוֹדוּלִים** **demodularization**

בתכן תוכנה – תהליך איחוד של מודולי תוכנה, שיש ביניהם קשר, הנעשה בדרך כלל כדי למטב את ביצועי המערכת.

20.03.27 **דְּחִיסָה מַטָּה** **downward compression**

בתכן תוכנה – צורת איחוד מודולים שבה מודול עילי מועתק לתוך הגוף של מודול תחתי.

20.03.28 **דְּחִיסָה צִדִּית** **lateral compression**

בתכן תוכנה – צורת איחוד מודולים שבה כמה מודולים המבוצעים בזה אחר זה, מאוחדים לתוך מודול יחיד.

20.03.29 **דְּחִיסָה מַעְלָה** **upward compression**

בתכן תוכנה – צורת איחוד מודולים שבה מודול תחתי מועתק למקום המתאים בתוך הגוף של מודול עילי.

20.03.30 **פֵּרוּק לְמוֹדוּלִים** **modular decomposition**

**מוֹדוּלַרִיזַצְיָה** **modularization**

תהליך פירוק של מערכת לרכיבים כדי לאפשר תכן ופיתוח.

**הערה**:

פירוק למודולים הוא אלמנט של תכנות מודולרי.

20.03.31 **פֵּרוּק תִּפְקוּדִי** **functional decomposition**

סוג של פירוק מודולרי, שבו מפרקים מערכת לרכיבים המתאימים לתפקודים ולתת-תפקודים של המערכת.

20.03.32 **פֵּרוּק מִדְרָגִי** **hierarchical decomposition**

סוג של פירוק מודולרי, שבו מפרקים מערכת לרכיבים מדרגיים באמצעות סדרת פעולות עידון מעלה-מטה.

20.03.33 **פֵּרוּק לְגוֹרְמִים** **Factoring**

תהליך פירוק מדרגי של מערכת, שבו מוציאים פונקצייה מתוך המודול שלה ומציבים אותה במודול עצמאי.

20.03.34 **תַּרְשִׁים תְּצוּרָה** **configuration diagram**

תרשים מלבנים של תצורה.

20.03.35 **זִהוּי תְּצוּרָה** **configuration identification**

הברירה והזיהוי של פריטי תצורה לצורך בקרת תצורה.

20.03.36 **נִהוּל תְּצוּרָה** **configuration management**

**CM** [abb]

נוהל של הכוונה ומעקב טכניים ומינהליים החל על זיהוי תצורה, בקרת תצורה, דיווח מיצב תצורה **(configuration status)** וביקורת תצורה.

20.03.37 **תְּצוּרָה כְּלָלִית** **form, fit and function**

בניהול תצורה – תצורה הכוללת את האופיינים הפיזיים והתפקודיים של יחידה תפקודית, אך מוציאה מכלל זה אופיינים של האלמנטים שמהם היחידה התפקודית מורכבת.

**הערה**:

תרגום מילולי של המונח האנגלי: צורה, הלימות ותפקוד.

20.03.38 **בַּקָּרַת מִנְשָׂק** **interface control**

בניהול תצורה – התהליך שבו:

א - מזהים את כל האופיינים התפקודיים והפיזיים המשמעותיים למישוק בין כמה פריטי תצורה, וכן

ב - מוודאים שהשינויים המוצעים באופיינים אלה נשקלים ומאושרים לפני מימושם.

20.03.39 **דִּווּחַ מִצַּב הַתְּצוּרָה** **configuration status accounting**

אלמנט בניהול תצורה, הכולל רישום ומסירה של מידע הנחוץ לניהול תצורה אפקטיבי.

**הערה**:

מידע זה כולל רשימות של זיהויי התצורה המאושרים ושל מִיצב השינויים המאושרים בתצורה ומיצב המימוש של שינויים מאושרים.

20.03.40 **קֻפְסָה שְׁחוֹרָה** **black box**

יחידה תפקודית שהקלט, הפלט והתפקודים שלה ידועים, אך פרטי המימוש והתכולה שלה אינם מובאים בחשבון.

20.03.41 **קֻפְסָה שְׁקוּפָה** **glass box**

**קֻפְסָה לְבָנָה** **white box**

מערכת או רכיב שהתכולה הפנימית או המימוש שלהם ידועים.

20.03.42 **מַטְרִיצַת עֲקִיבוּת** **traceability matrix**

מטריצה שבה נרשמת היוחסה בין כמה מוצרים של תהליך הפיתוח.

**דוגמה**:

מטריצה שבה נרשמת היוחסה בין הדרישות לבין התכן של רכיב תוכנה נתון.

20.03.43 **מַטְרִיצַת תְּנוּעוֹת** **transaction matrix**

מטריצה המזהה בקשות אפשריות לגישה אל מסד הנתונים ומייחסת כל בקשה לקטגוריות מידע או לאלמנטים במסד הנתונים.

20.03.44 **הִתְנַגְּשׁוּת סְדָרִים** **order clash**

בתכן תוכנה – התנגשות מבנית, שבה סדר המיון של הנתונים הדרוש לתוכנית, שונה מסדר המיון שבו הנתונים מנוהלים.

20.03.45 **הִתְנַגְּשׁוּת מִבְנִים** **structure clash**

בתכן תוכנה – מצב שבו מודול חייב לעסוק בכמה קבוצות נתונים שמבני הנתונים שלהן אינם מתיישבים.

20.03.46 **מֻנְחֵה פְּקֻדּוֹת** **command-driven**

מתייחס למערכת או לאופן פעולה, שבהם המשתמש מנחה את המערכת באמצעות פקודות.

20.03.47 **מֻנְחֵה תַּפְרִיטִים** **menu-driven**

מתייחס למערכת או לאופן פעולה, שבהם המשתמש מנחה את המערכת באמצעות ברירת חלופות מתפריט.

20.03.48 **מַעֲקַף תַּפְרִיט** **menu by-pass**

במערכת מונחית תפריטים – מנגנון המאפשר למשתמשים מתקדמים לבצע תפקודים באופן מונחה פקודות ללא צורך בבחירת אפשרויות מהתפריטים.

20.03.49 **תּוֹאֵם מַטָּה** **downward compatible**

מתייחס ליוחסה שבין גרסאות של יחידות תפקודיות או של מוצריהן, המאפשרת שימוש בגִרסה מאוחרת במקום בגִרסה מוקדמת.

**דוגמה**:

חבילת תוכנה היכולה להשתמש בקבצים שנוצרו על ידי גרסה קודמת, נחשבת תוכנה תואמת-מטה, ואילו הקובץ הישן עצמו נחשב תואם-מעלה.

20.03.50 **תּוֹאֵם מַעְלָה** **upward compatible**

מתייחס ליוחסה שבין גרסאות של יחידות תפקודיות או של מוצריהן, המאפשרת שימוש בגִרסה מוקדמת במקום בגִרסה מאוחרת.

**דוגמה**:

קובץ שנוצר על ידי גִרסה מוקדמת של תוכנה ושניתן להשתמש בו בגִרסה מאוחרת נחשב תואם-מעלה, ואילו הגִרסה המאוחרת של התוכנה נחשבת תואמת-מטה.

20.03.51 **תַּסְקִיר תֶּכֶן** **design review**

תהליך או פגישה שבמהלכם מוצג התכן של מערכת או של יחידה תפקודית לפני צוות הפרויקט, מנהלים, משתמשים, לקוחות או צדדים מעוניינים אחרים, למתן הערות: או לאישור.

20.03.52 **תַּסְקִיר תֶּכֶן-קֶדְמִי** **preliminary design review**

**PDR** [abb]

תסקיר הנערך לכל רכיב של חומרה או תוכנה כדי להעריך את ההתקדמות, את הדיוק הטכני ואת התשובות לסיכונים של גישת התכן שנבררה עבור פריטי תצורה אחד או יותר, כדי לקבוע את המתיישבות של כל תכן עם הדרישות של פריט התצורה, כדי להעריך את רמת ההגדרה ולאמוד את הסיכון הטכני בשיטות הייצור ובתהליכי הייצור שנבררו, כדי לקבוע את הקיום ואת המתיישבות של המִנשקים הפיזיים והתפקודיים בין פריטי התצורה ופריטים אחרים של ציוד, של מתקנים, של תוכנה ושל כוח אדם, ואם הדבר ישים – כדי להעריך את התיעוד התפעולי ואת תיעוד התמיכה הקִדָמיים.

20.03.53 **תַּסְקִיר תֶּכֶן-מַעֲרֶכֶת** **system design review**

**SDR** [abb]

תסקיר הנערך כדי להעריך את אופן הקצאת דרישות המערכת לפריטי תצורה, את תהליך הנדסת המערכות ששימש להקצאה, את התכנון ההנדסי של שלב המאמץ הבא, את שיקולי הייצור ואת תכנון הנדסת הייצור.

20.03.54 **תַּסְקִיר תֶּכֶן-קְרִיטִי** **critical design review**

**CDR** [abb]

תסקיר התכן של כל רכיב חומרה או של כל רכיב תוכנה, הנערך כדי לאמת שהתכן המפורט של פריט תצורה אחד או יותר ממלא אחר הדרישות שפורטו.

**הערה**:

התסקיר עשוי לכלול:

- אימות המתיישבות בין פריטי התצורה ופריטים אחרים של ציוד, של מתקנים, של תוכנה ושל כוח אדם;
- הערכת תוצאות ניתוחים של אפשרויות הייצור של המוצר;
- הערכת התכנון הקדמי של הבדיקות;
- הערכת נאותות התיעוד התפעולי ותיעוד התמיכה הקְדָמיים.

20.03.55 **בִּיקֹרֶת נוֹקֶבֶת מֻבְנִית** **structured walk-through**

**בִּיקֹרֶת נוֹקֶבֶת** **walk-through**

בדיקה שיטתית של הדרישות, של התכן או של המימוש של מערכת או של חלק ממנה על ידי בני-סמכא.

**דוגמה**:

טכניקת ניתוח שבה מתכנן או מתכנת מוביל חברים בצוות הפיתוח וצדדים מעוניינים אחרים דרך קטע של תיעוד או קוד, והמשתתפים שואלים שאלות ומעירים הערות:

על שגיאות אפשריות, על הפרות של תקני פיתוח ועל בעיות אחרות..

20.03.56 **צִנּוֹר** **pipe**

בתכנות מחשבים -טכניקה שבה נתונים של תהליך בתוכנית מסוימת מועברים לתהליך אחר.

**הערה**:

הסימן המקובל של צינור הוא קו אנכי.

20.03.57 **הַצְנָרָה** **pipelining**

שימוש בצנרת.

20.03.58 **אֲרִיזָה** **packaging**

בתכן תוכנה - הקצאה של מודולים למקטעים שיטופלו כיחידות פיזיות נבדלות לביצוע במחשב.

20.03.59 **תִּכְלֵל** **integrate**

לקח נתונים במגוון תסדירים שונים וממגוון מקורות שונים ומיזג אותם לתסדיר יחיד.

20.03.60 **תַּשְׁתִּית מְתֻכְלֶלֶת** **integrated infrastructure**

מצע שעליו מריצים יישומים והמאפשר להם לחלוק תפקודיות.

20.03.61 **רְגִישׁוּת בְּרָמַת הַשָּׂדֶה** **field-level sensitivity**

היכולת של תוכנית יישום לגשת לנתונים ברמת שדה.

20.03.62 **פִּתּוּחַ מְבוּסַּס רְכִיבִים** **component-based development**

פיתוח תוכנה המשלב רכיבי תוכנה קיימים.

20.03.63 **תְּצוּרָה מֻדְמֵית** **simulated configuration**

תצורה שלא ממומשת בפועל אלא משמשת רק לצורך אומדן עלויות אילו נוצרה.

20.03.64 **בְּסֵס מֵחָדָשׁ** **rebase** (v)

הגדיר מחדש קו-בסיס לפיתוח תוכנה.

## 20.04 - מימוש - Implementation

20.04.01 **הַנְדָּסַת תָּכְנָה בְּעֶזְרַת מַחְשֵׁב** **computer-aided software engineering**
**CASE** [abb]

שימוש בערכת כלים או תוכניות לעזרה בפיתוח יישומים מרוכבים.

20.04.02 **תִּכְלוּל מַעֲרָכוֹת** **system integration**

הרכבה מודרגת (שלב-שלב) של רכיבי המערכת למערכת שלמה אחת

**הערה**: מקובל בציבור "שילוב מערכות".

20.04.03 **מַעֲרֶכֶת מַעֲבָר** **change-over system**

מערכת עיבוד מידע זמנית, המשמשת להקלת המעבר ממערכת פועלת אחת אל המערכת הבאה במקומה.

20.04.04 **הֶחְלֵף** **cutover**

העברה של תפקודי מערכת אחת למערכת אחרת הבאה במקומה, ברגע נתון.

20.04.05 **רִיצָה בְּמַקְבִּיל** **parallel run**
**הַפְעָלָה בְּמַקְבִּיל**

הפעלה של שתי מערכות עיבוד מידע, אחת הפועלת ומחליפתה המיועדת, עם אותו יישום ועם אותם נתוני מקור, לשם השוואה ובדיקת האמינות של המערכת המחליפה.

20.04.06 **תְּצוּרַת פִּתּוּחַ** **developmental configuration**

בניהול תצורה - התוכנה והתיעוד הטכני המסומך אליה, המגדירים את התצורה המתפתחת של פריט תצורה של תוכנת מחשב בעת הפיתוח.

**הערה**:

תצורת הפיתוח נמצאת בשליטת המפַתח, ולכן אינה נקראת קו-בסיס.

20.04.07 **תִּכְנוּת לֹא-אָנוֹכִי** **egoless programming**

טכניקת פיתוח תוכנה המבוססת על תפיסת צוות, המטילה את האחריות לפיתוח על צוות ולא אישית על כל מפַתח.

**הערה**: המטרה - השיטה היא למנוע מתוכניתנים הזדהות אישית מדי עם עבודתם עד כדי שיבוש יכולתם לבצע הערכה אובייקטיבית.

20.04.08 **קִדּוּד מִתְחָרֶה** **software diversity**
**קִדּוּד כָּפוּל** **dual coding**

טכניקת פיתוח תוכנה שבה מפותחות כמה גרסאות תוכנה זהות מבחינה תפקודית לפי אותו מפרט, על ידי תוכניתנים שונים או על ידי צוותי תכנות שונים, מתוך כוונה לגלות שגיאות, להגדיל אמינות, להוסיף תיעוד או להפחית את הסבירות ששגיאות תכנות או הידור ישפיעו על התוצאות הסופיות.

20.04.09 **תַּסְקִיר קוֹד** **code review**

תסקיר של קוד המקור של תוכנית מחשב לצורך הערות או אישור.

20.04.10 **סִפְרִיַּת פִּתּוּחַ תָּכְנָה** **software development library**
**סִפְרִיַּת פְּרוֹיֶקְט** **project library**
**סִפְרִיַּת תְּמִיכָה בְּתָכְנָה** **program support library**

ספריית תוכנה המכילה מידע קריא-מחשב ומידע קריא-אנוש, המשמעותי למאמץ פיתוחה תוכנה.

20.04.11 **נְתוּנֵי מַעֲרֶכֶת** **system data**

נתוני מערכת המחשב שבה מתבצע היישום.

20.04.12 **דוּ-קִיּוּם** **coexistence**

במהלך ההגירה מגרסת תוכנה אחת לגרסה אחרת, המצב שבו שתי הגרסאות קיימות זו בצד זו באותה מערכת לפרק זמן מסוים.

## 20.05 - הבטחת איכות - Quality Assurance

20.05.01 **אֵיכוּת** **quality**

בפיתוח מערכות - המידה שבה מערכת, רכיב או עיבוד, עונים על דרישות לקוח מוגדרות.

20.05.02 **הַבְטָחַת אֵיכוּת** **quality assurance**

**QA** [abb]

פעילויות שיטתיות מתוכננות, הנדרשות כדי להבטיח איכות של רכיב או של מערכת.

20.05.03 **בַּקָּרַת אֵיכוּת** **quality control**

**QC** [abb]

פעילויות המשמשות להערכת מילוי דרישות הבטחת האיכות.

**הערה**:

בבקרת איכות משולבות טכניקות ופעילויות תפעוליות, המיועדות לניטור תהליך הפיתוח ולצמצום הגורמים לביצועים שאינם משביעי רצון בכל השלבים של מחזור האיכות, כדי להשיג אפקטיביות כלכלית.

20.05.04 **תְּכוּנַת אֵיכוּת** **quality attribute**

**גּוֹרֵם אֵיכוּת** **quality factor**

מאפיין או אופיין המשפיעים על איכות של פריט.

20.05.05 **מֶטְרִיקַת אֵיכוּת** **quality metric**

מדד כמותי (מטריקה) המציין את הדרגה שבה פריט ניחן בתכונת איכות נתונה.

**הערה**:

ראו גם: **מטריקת איכות תוכנה**.

20.05.06 **מֶטְרִיקָה** **metric**

מדד כמותי של הדרגה שבה מערכת, רכיב או תהליך, הם בעלי תוכנה נתונה.

20.05.07 **עֶקִיבוּת** **traceability**

הדרגה שבה אפשר לכונן יוחסה בין כמה מוצרים של תהליך הפיתוח, במיוחד מוצרים שהיוחסה ביניהם היא יוחסת קודם-יורש או ממונה-כפוף, כגון דרגת ההתאמה בין הדרישות לבין התכן של רכיב תוכנה נתון.

**הערה**:

עקיבות עשויה להתייחס למקורם של חומרים וחלקים, להיסטוריית העיבוד או להפצה של מוצר, היא עשויה לייחס ציוד מדידה לתקנים לאומיים או בין-לאומיים, או לחומרי ייחוס, או לייחס חישובים ונתונים שהופקו במהלך מחזור האיכות ולעתים מתייחסת העקיבות לדרישות האיכות.

20.05.08 **קְרִיטֶרְיוֹן קַבָּלָה** **acceptance criterion**

תנאי שיחידה תפקודית חייבת לעמוד בו כדי להתקבל על ידי משתמש, על ידי לקוח או על ידי גורם מורשה אחר.

**דוגמה**:

הדרגה שבה מערכת חייבת לעמוד בקריטריונים לבדיקה.

20.05.09 **פְּרוֹפִיל מַעֲרֶכֶת** **system profile**

אוסף מדידות המשמש להערכת ביצועי מחשב, המתאר את החלק היחסי של הזמן שבו תפוס כל אחד מהמשאבים העיקריים של מערכת מחשב, מתוך הזמן שאותו משאב זמין.

20.05.10 **כְּלִילוּת מַעֲרֶכֶת** **system integrity**

בעיבוד נתונים - מצב של ביטחון מלא בכך, שמערכת עיבוד הנתונים מבוססת על נכונות לוגית של החומרה והתוכנה המממשות את מנגנוני ההגנה שלה, ועל כלילות הנתונים, בכל תנאי.

20.05.11 **הַתְעָדָה** **certification**

בפיתוח מערכות - נוהל של מתן ערובה בכתב, שמערכת או רכיב-מערכת עומדים בדרישות מוגדרות וקבילים לשימוש תפעולי.

20.05.12 **כְּשִׁירוּת** **qualification**

**תַּהֲלִיך הוֹכָחַת כְּשִׁירוּת** **qualification process**

התהליך הקובע אם מערכת או רכיב עומדים בדרישות מוגדרות ומתאימים לשימוש תפעולי.

20.05.13 **תַּסְקִיר כְּשִׁירוּת פּוֹרְמָלִי** **formal qualification review**

**FQR** [abb]

הבדיקה או תהליך הניתוח, שבהם מאמתים כי קבוצת פריטי תצורה המרכיבים מערכת, עומדת בדרישות ביצועים חוזיות.

20.05.14 **תַּסְקִיר מוּכָנוּת לִבְדִיקָה** **test readiness review**

**TRR** [abb]

תסקיר של פריטי תצורה, שמטרתו להעריך תוצאות בדיקה ראשוניות, לאמת שנוהלי הבדיקה של כל פריט תצורה הושלמו ועומדים בתוכניות ובמפרטי הבדיקה וכן כדי לאמת שהפרויקט מוכן לבדיקה פורמלית של פריטי התצורה.

20.05.15 **הַעֲרָכַת בִּצּוּעֵי מַחְשֵׁב** **computer performance evaluation**

**CPE** [abb]

שיטה ללימוד ולמדידה של ביצועי מחשב ולחקר הדרכים לשיפור ביצועים אלה.

20.05.16 **עֹמֶס עֲבוֹדָה** **workload**

תמהיל משימות המורץ בדרך כלל במערכת מחשב נתונה, ושהאופיינים העיקריים שלו כוללים דרישות קלט/פלט, כמות חישובים וסוגיהם ומשאבי מחשב דרושים.

20.05.17 **מוֹדֵל עֹמֶס עֲבוֹדָה** **workload model**

מודל המשמש להערכת ביצועי מחשב והמתאר את ניצול המשאבים ומדדי הביצועים של עומסי עבודה צפויים או ממשיים במערכת מחשב.

20.05.18 **בְּחִינַת קוֹד** **code inspection**

טכניקת ניתוח סטטי, המסתמכת על בחינת תוכניות מחשב לשם גילוי שגיאות, הפרות תקני פיתוח ובעיות אחרות.

20.05.19 **בְּחִינַת תֶּכֶן** **design inspection**

טכניקת ניתוח סטטי המסתמכת על בחינת המוצרים שבפיתוח לשם גילוי שגיאות, הפרות תקני פיתוח ובעיות אחרות.

20.05.20 **מִבְדַּק תְּצוּרָה תִּפְקוּדִי** **functional configuration audit**

**FCA** [abb]

מבדק הנערך כדי לאמת שהפיתוח של פריט תצורה הושלם באופן משביע רצון, שהפריט השיג את האופיינים הביצועיים והתפקודיים שפורטו בזיהוי התצורה ושהתיעוד התפעולי ותיעוד התמיכה של הפריט שלמים ומספקים.

20.05.21 **מִבְדַּק תְּצוּרָה פִּיזִי** **physical configuration audit**

**PCA** [abb]

מבדק הנערך כדי לאמת שפריט תצורה, כפי שנבנה, עומד בדרישות התיעוד הטכני המגדיר אותו.

20.05.22 **נִתּוּחַ דִּינָמִי** **dynamic analysis**

תהליך הערכה של מערכת או רכיב, המבוססת על התנהגותם בעת הביצוע.

20.05.23 **נִתּוּח סְטָטִי** **static analysis**

תהליך הערכה של יחידה תפקודית, המבוססת על הצורה, על המבנה, על התוכן או על התיעוד שלה.

20.05.24 **בְּדִיקָה** **testing**

התהליך שבו מופעלים מערכת, רכיב או פריט תוכנה, בתנאים מוגדרים, תוך תצפית או רישום של התוצאות, כדי להעריך היבטים ומאפיינים מסוימים שלהם וכדי לגלות הבדלים בין התוצאות בפועל לבין התוצאות הנדרשות.

20.05.25 **בְּדִיקַת אַלְפָּא** **alpha testing**

בדיקה הנערכת על ידי הארגון המפתח לפני שחרור המוצר למשתמשים פוטנציאליים.

20.05.26 **בְּדִיקַת בֵּיתָא** **beta testing**

בדיקת לא-פורמלית הנערכת על ידי משתמשים פוטנציאליים לפני השחרור הרשמי של המוצר.

20.05.27 **בְּדִיקָה תִּפְקוּדִית** **functional testing**

**בְּדִיקַת קֻפְסָה שְׁחוֹרָה** **black-box testing**

בדיקה הנערכת כדי להעריך את עמידת המערכת או הרכיב בדרישות תפקודיות מוגדרות.

**הערה**:

בדיקה תפקודית מתעלמת מהמנגנון הפנימי של המערכת או של הרכיב ומתמקדת בלעדית בפלט המופק במענה לקלט שנברר ולתנאי הביצוע.

20.05.28 **בְּדִיקַת פִּיתּוּחַ** **developmental testing**

בדיקה פורמלית או לא-פורמלית, הנערכת במהלך הפיתוח של מערכת או של רכיב, בדרך כלל על ידי המְפַתח, בסביבת הפיתוח.

20.05.29 **בְּדִיקָה פּוֹרְמָלִית** **formal testing**

בדיקה הנערכת על פי תוכניות הבדיקה ונוהלי הבדיקה שנסקרו ואושרו על ידי לקוח, על ידי משתמש או על ידי דרג ניהולי מוסמך.

20.05.30 **בְּדִיקָה לֹא-פּוֹרְמָלִית** **informal testing**

בדיקה הנערכת בשיטות פחותות משיטות הבדיקה פורמליות, כגון שימוש בתוכניות בדיקה ובנוהלי בדיקה שלא נסקרו ושלא אושרו על ידי לקוח.

20.05.31 **בְּדִיקַת תִּכְלוּל** **integration test**

בדיקה שבה משלבים רכיבי חומרה או רכיבי תוכנה, או את שניהם יחד, ובודקים אותם כדי להעריך את ההידוד (interaction) ביניהם וכדי להבטיח את תפקודם הנכון במערכת השלמה.

20.05.32 **בְּדִיקַת מִנְשָׁק** **interface testing**

בדיקה הנערכת כדי לקבוע אם מערכות או רכיבים עומדים במפרטי המִנשק שלהם.

20.05.33 **בְּדִיקַת גַּב-אֶל-גַּב** **back-to-back testing**

בדיקה שבה מריצים כמה גִרסאות של תוכנית עם אותם קלטים, משווים את הפלטים ומנתחים שגיאות כשמתגלות אי-התאמות.

20.05.34 **בְּדִיקַת לוּלָאָה** **loopback testing**

בדיקה שבה אותות או נתונים מהתקן בדיקה הם קלט למערכת או לרכיב, והתוצאות מוחזרות להתקן הבדיקה לשם מדידה או השוואה.

20.05.35 **בְּדִיקַת מוּטַצְיָה** **mutation testing**

מתודולוגיית-בדיקה שבה משתמשים באותם מקרי-בדיקה (test cases) לביצוע כמה מוטציות של תוכנית כדי להעריך את היכולת של מקרי-הבדיקה לגלות הבדלים במוטציות של התוכנית.

20.05.36 **בְּדִיקַת תִּפְעוּל** **operation testing**
**operator testing**

בדיקה הנערכת כדי להעריך יחידה תפקודית כשהיא פועלת בסביבתה המיועדת.

20.05.37 **בְּדִיקַת נָתִיב** **path testing**

בדיקה שנתכנה כדי לעקוב אחר מעקובות ביצוע בתוכנית מחשב - כולן או מבחר מתוכן.

20.05.38 **בְּדִיקַת בִּצּוּעִים** **performance testing**

בדיקה הנערכת כדי להעריך את עמידתה של יחידה תפקודית בדרישות ביצועים מוגדרות, או כדי להשוות ביצועים של שני מימושים או של שתי תוכניות.

20.05.39 **בְּדִיקַת כְּשִׁירוּת** qualification testing

בדיקה הנערכת כדי לקבוע אם מערכת או רכיב מתאימים לשימוש תפעולי.

20.05.40 בְּדִיקַת תְּסוּגָה **regression testing**

בדיקה-חוזרת סלקטיבית של מערכת או של רכיב, הנערכת כדי לוודא שתיקונים שבוצעו לא הביאו לתוצאים לא רצויים ושהמערכת או הרכיב עומדים עדיין בדרישות שפורטו.

20.05.41 **בְּדִיקַת מִשְׁפָּטִים** statement testing

בדיקה שנתכנה לבצע כל משפט בתוכנית מחשב.

20.05.42 בְּדִיקַת מַאֲמָץ **stress testing**

בדיקה הנערכת כדי להעריך מערכת או רכיב על גבול הדרישות המוגדרות שלהם או מעבר לכך.

20.05.43 **בְּדִיקַת מִבְנֶה** **structural testing**
**בְּדִיקַת קֻפְסָה שְׁקוּפָה** **glass-box testing**
**בְּדִיקַת קֻפְסָה לְבָנָה** **white-box testing**

בדיקה המביאה בחשבון את המנגנון הפנימי של מערכת או של רכיב.

**הערה**:

טיפוסי בדיקות מבנה כוללים: בדיקת הסתעפות, בדיקת נתיב ובדיקת משפטים.

20.05.44 **בְּדִיקַת מַעֲרֶכֶת** **system testing**

בדיקה הנערכת על מערכת משולבת שלמה כדי להעריך את עמידתה בדרישות מוגדרות.

20.05.45 **בְּדִיקַת יְחִידָה** **unit testing**

בדיקת יחידת חומרה, או תוכנית או מודול, כדי להבטיח שאין בהם טעויות בניתוח או בתכנות.

20.05.46 **בִּדּוּק יְדָנִי** **desk checking**

טכניקת ניתוח שבה מבוצעת הדמיה ידנית של קטעי קוד-מקור ותוצאות הבדיקה או תיעוד אחר נבחנים חזותית, בדרך כלל על ידי האדם שהפיק אותם, כדי לזהות שגיאות, הפרות של תקני פיתוח או בעיות אחרות.

**הערה**:

בידוק ידני מכונה גם "ריצה יבשה".

20.05.47 **בְּדִיקַת קַבָּלָה** **acceptance test**

בדיקה של מערכת או של יחידה תפקודית, המבוצעת בדרך כלל על ידי הלקוח בחצריו, לאחר ההתקנה ובשיתוף הספק, כדי להבטיח שנתקיימו דרישות החוזה.

20.05.48 **בֹּחַן עַל פִּי נְקֻדַּת מִדּוּד** **benchmark test**

**בּוֹחַן בִּצּוּעִים** **benchmark**

בדיקה המיועדת להערכת הביצועים של חומרה ושל תוכנה בתצורה נתונה, על פי אמות מידה או תקנים מוגדרים.

**הערה**: הערכת הביצועים נעשית למשל באמצעות:

1. קבוצה מייצגת של תוכניות ויישומים, היוצרים מצבים ותנאים המשמשים להערכת ביצועים;
2. תוכנית מחשב שנתכנה במיוחד לספק מדידות של מערכת הפעלה מסוימת או של יישום מסוים;
3. מוצר ידוע, שביצועיו מוכרים ומקובלים בקרב משתמשים, ושאפשר להשוות אליהם את ביצועי המוצר הנבדק;
4. קבוצת קריטריוני ביצועים שהמוצר אמור לעמוד בהם.

20.05.49 **אֲמִיתוּת** **verification**

**בְּדִיקַת אֲמִיתוּת** **verification test**

בדיקה הנערכת כדי להוכיח שמערכת עומדת בכל הדרישות המוגדרות בשלב מסוים של הפיתוח.

20.05.50 **תִּקּוּף** **validation**

**בְּדִיקַת תִּקּוּף** **validation test**

בדיקה הנערכת כדי לקבוע אם מערכת שמומשה עומדת בדרישות מוגדרות.

20.05.51 **בְּדִיקַת יחידה** **unit test**

בדיקת יחידות חומרה, או בדיקת תוכניות או מודולים, כדי להבטיח שאין טעויות בניתוח או בתכנות.

20.05.52 **בְּדִיקַת שְׁמִישׁוּת** **usability test**

**fitness-for-use test**

בדיקה הנערכת כדי לקבוע שמערכת ממומשת עומדת בדרישות המוגדרות של המשתמשים.

20.05.53 **אֵרוּעַ בדיקת תָּכְנָה** **Software test incident**

**אֵרוּעַ** **incident**

מקרה שהתרחש במהלך בדיקת תוכנה והדורש חקירה.

20.05.54 **אַרְכִּיטֶקְטוּרַת עֶרְכַּת בְּדִיקָה** **test set architecture**

היוחסות המקוּנָנות בין קבוצות של מקרי בדיקה, המשפיעות ישירות על הפירוק המדרגי של מטרות הבדיקה.

20.05.55 **מְחוֹלֵל בְּדִיקוֹת** **test generator**

**מְחוֹלֵל נְתוּנֵי בְּדִיקָה** **test data generator**

**test generator**

כלי תוכנה המקבל כקלט קוד מקור, קריטריונים לבדיקה, מפרטים, או הגדרות של מבני נתונים, משתמש בהם כדי לחולל נתוני קלט לבדיקה ולעתים קובע את התוצאות הצפויות.

20.05.56 **הֶנֵּעַ בְּדִיקָה** **test driver**

**test harness**

חומרה או תוכנה המפעילות יחידה תפקודית שבבדיקה והעשויות גם לספק לה קלט לבדיקה, לשלוט בביצועיה ולנטרם, ולדווח את תוצאותיה.

20.05.57 **מַעֲרֶכֶת אִמּוּת מְמֻחְשֶׁבֶת** **automated verification system**

כל כלי תוכנה הממחשב את תהליך האימות, כולו או חלקו.

20.05.58 **מַשְׁוָן** **comparator**

כלי תוכנה המשווה שתי תוכניות מחשב, שתי קבוצות נתונים או שני קבצים, כדי לזהות את השווה או את השונה שבהם.

**הערה:**

עצמים טיפוסיים להשוואה הם גרסאות זהות של קוד מקור, קוד יעד, קובצי מסד-נתונים או תוצאות בדיקה.

20.05.59 **נוֹסַח אַחֵר** **variant**

בסבולת תקלות - גרסה של תוכנית שהיא תוצאת השימוש בקידוד כפול.

20.05.60 **מוּטַצְיָה שֶׁל תָּכְנִית** **program mutation**

**מוּטַצְיָה** **mutation**

גרסה של תוכנית מחשב שבה שונתה הגרסה המקורית כדי להעריך את היכולת של מקרי בדיקה לגלות את השינוי.

20.05.61 **בִּצּוּעַ חוֹזֵר** **reversible execution**

**backward execution**

**replay**

טכניקת ניפוי שגיאות שבה נרשמת היסטוריית הביצוע של תוכנית ולאחר מכן מבוצעת התוכנית מחדש, קדימה או אחורה, בפיקוח המשתמש.

20.05.62 **עִקְבַת תַּקָּלוֹת** **fault trace**

רשימת תקלות שהופקה בעזרת מוניטור, המשקפת את סדרת המצבים שקדמה ממש להיקרות התקלות.

20.05.63 **סַף תַּקָּלוֹת** **fault threshold**

גבול מוגדר מראש למספר תקלות מסוג מסוים, שאם עוברים אותו, יש לנקוט אמצעים מתאימים.

**הערה:**

אמצעים מתאימים הם למשל הודעה למפעילים, הרצת תוכניות אבחון, או שינוי התצורה לסילוק יחידה לקויה.

20.05.64 **סַף קֶצֶב תַּקָּלוֹת** **fault-rate threshold**

סף תקלות המבוטא על ידי מספר התקלות במשך זמן מוגדר.

20.05.65 **מֶטְרִיקַת אֵיכוּת** **software quality metric**

מדד כמותי (מטריקה) של איכות תוכנה, המחושב על ידי פונקציה שהקלט שלה הם נתוני התוכנה והפלט שלה הוא ערך מספרי יחיד המבטא את דרגת האיכות של התוכנה.

20.05.66 **נְקֻדַּת מִדּוּד** **Benchmark**

אמות מידה או תקנים מוגדרים, המשמשים בבדיקה המיעודת להערכת הביצועים של חומרה ושל תוכנה בתצורה נתונה.

## 20.06 - תפעול ותחזוקה - Operation and Maintenance

20.06.01 **תַּחְזוּקָה** **maintenance**

פעולות המבוצעות לשם שמירת יחידה תפקודית במצב תפעולי תקין או לשם החזרתה למצב זה.

20.06.02 **תַּחְזוּקַת מַעֲרֶכֶת** **system maintenance**

הסגלה (modification) של מערכת הנעשית כדי לתקן תקלות, לשפר ביצועים, או להתאים את המערכת לסביבה או לדרישות שהשתנו.

20.06.03 **תַּחְזוּקָה מְסַגֶּלֶת** **adaptive maintenance**

תחזוקה המבוצעת כדי להתאים יחידה תפקודית לסביבה שהשתנתה.

20.06.04 **תַּחְזוּקָה מְתַקֶּנֶת** **corrective maintenance**

תחזוקה המבוצעת לאחר התרחשות כשל או גילוי תקלה, כדי להחזיר יחידה תפקודית למצב תפקודי תקין.

20.06.05 **תַּחְזוּקָה מְשַׁפֶּרֶת** **perfective maintenance**

תחזוקה המבוצעת כדי לשפר את הביצועים, את התחזוקתיות או תכונות אחרות של תוכנית מחשב.

20.06.06 **תַּחְזוּקָה מוֹנַעַת** **preventive maintenance**

תחזוקה המבוצעת במרווחי זמן מוגדרים מראש או על פי קריטריונים מפורטים, כדי להפחית את ההסתברות לתקלה או כדי למנוע הידרדרות בביצועים של יחידה תפקודית.

20.06.07 **בְּדוּק שִׁחְרוּר** **checkout**

בדיקה הנערכת בסביבה התפעולית או בסביבת התמיכה של מוצר כדי לוודא שהמוצר מתפקד כנדרש לאחר התקנתו.

20.06.08 **יַתִּירוּת** **redundancy**

בסבולת תקלות - קיום של רכיבים יתירים במערכת, המבצעים תפקודים זהים או דומים לתפקודים של אלמנטים אחרים, במטרה למנוע כשלים או להתאושש מהם.

**דוגמות:**

שימוש ברכיבים תפקודיים כפולים, הוספת סיבית זוגיות.

**הערה**:

יתירות משמשת בעיקר לשיפור האמינות והזמינות.

20.06.09 **יַתִּירוּת פְּעִילָה** **active redundancy**

בסבולת תקלות - השימוש ביחידות תפקודיות יתירות הפעילות בו-זמנית, כדי למנוע תקלות או לאפשר התאוששות מהן.

20.06.10 **יַתִּירוּת בַּעֲתוּדָה** **standby redundancy**

בסבולת תקלות - השימוש ביחידות תפקודיות יתירות שאינן פעילות עד שמתרחשת תקלה באלמנט הפעיל.

20.06.11 **יַתִּירוּת אֶחִידָה** **homogeneous redundance**

בסבולת תקלות - מימוש של אותה פונקצייה באמצעים זהים.

**דוגמה**:

שימוש בשני מעבדים זהים.

20.06.12 **דִּשְׁדּוּשׁ** **Thrashing**

מצב שבו מערכת מחשב משקיעה את כל משאביה או את מרביתם בפעולות תקורה כגון חילוף נתונים בין זיכרון ראשי לבין זיכרון עזר, במקום בתפקודי החישוב הנדרשים.

20.06.13 **מִכְשׁוּר** **Instrumentation**

הֶתקנים המיתוספים בחומרה או הוראות המוכנסות לתוכנה כדי לנטר את התפעול של מערכת או של רכיב.

20.06.14 **שִׁדּוּד מַעֲרָכוֹת** **system sizing**

20.06.15 **שִׁדּוּד מַעְלָה** **Upsizing**

20.06.16 **שִׁדּוּד מַטָּה** **Downsizing**

שינוי גודל של מערכות על ידי הסרת חומרה ותוכנה כדי להתאימן לצרכים.

20.06.17 **שִׁדּוּד מֵחָדָשׁ** **Resizing**

שינוי פרמטרים, קואורדינטות או ממדים של אלמנטים על משטח תצוגה.

**דוגמה**:

שינוי גודל תמונה גרפית, כדי להתאימה למסגרת הטקסט.

20.06.18 **שִׁדּוּד הוֹלֵם** **right-sizing**

שינוי גודל התואם למטרה.

## 20.07 - תיעוד - Documantation

20.07.01 **תִּעוּד מַעֲרֶכֶת** **system documentation**

אוסף המסמכים המתארים את הדרישות, את היכולות, את המגבלות, את התכן, את התפעול ואת התחזוקה של מערכת.

20.07.02 **תֵּאוּר מַעֲרֶכֶת** **system description**

תיעוד הנובע מתכן מערכות המגדיר את הארגון, את האופיינים העיקריים ואת דרישות החומרה והתוכנה של המערכת.

20.07.03 **תִּעוּד מִשְׁתַּמֵּשׁ** **user documentation**

תיעוד המתאר כיצד יש להשתמש במערכת או ברכיב כדי להשיג תוצאות רצויות.

20.07.04 **מִסְמַךְ תְּכָן** **design document**

**תֵּאוּר תְּכָן** **design description**

**מִפְרַט תְּכָן** design specification

בפיתוח מערכות - מסמך המתאר את התכן של מערכת או של רכיב, והעשוי לכלול ארכיטקטורה, לוגיקת בקרה, מבני נתונים, תסדירי קלט/פלט, תיאורי מִנשקים ואלגוריתמים.

20.07.05 **מִפְרַט דְּרִישׁוֹת תָּכְנָה** **software requirements specification**

**SRS** [abb]

תיעוד הדרישות העיקריות (תפקודים, ביצועים, אילוצי תכן ותכונות) של תוכנה ושל מִנשקי החוץ שלה.

20.07.06 **מִפְרַט תָּכְנִית** **program specification**

מסמך המתאר את המבנה ואת התפקודים של תוכנית והמפורט דיו כדי לאפשר תכנות ותחזוקה.

20.07.07 **מִפְרַט תְּפְקוּדִי** **functional specification**

מסמך המפרט את התפקודים שמערכת או רכיב חייבים לבצע.

20.07.08 **מִפְרַט מִקְרֵה בְּדִיקָה** **test case specification**

**מִפְרַט בְּדִיקָה** **test specification**

מסמך המפרט את הקלט לבדיקה, את תנאי הביצוע ואת התוצאות הצפויות.

20.07.09 **קַו-בָּסִיס מֻקְצֶה** **allocated baseline**

בניהול תצורה - מפרטי התפקודיות התחיליים של פריט תצורה, שהוקצו ואושרו לפני הפיתוח.

20.07.10 **קַו-בָּסִיס לְפִתּוּחַ** **developmental baseline**

בניהול תצורה - המפרטים שבתוקף במועד מסוים לגבי מערכת הנמצאת בפיתוח והמשמשים כתצורת ייחוס לניתוח תוצאות של בדיקות הנערכות בתהליך הפיתוח.

20.07.11 **קַו-בָּסִיס תְּפְקוּדִי** **functional baseline**

בניהול תצורה - התיעוד הפורמלי של הדרישות התפקודיות של מערכת או של פריט תצורה.

20.07.12 **קַו-בָּסִיס לְמוּצָר** **product baseline**

בניהול תצורה - התיעוד הטכני המאושר התקף במשך שלבי הייצור, התפעול, התחזוקה והתמיכה הלוגיסטית במחזור החיים של הפרויקט.

**הערה:**

בתוכנה, התיעוד הטכני כולל רשימות של קוד מקור.

20.07.13 **מִסְמַךְ דְּרִישׁוֹת** **requirements document**
**מִפְרַט דְּרִישׁוֹת** **requirement specification**

מסמך המגדיר את הדרישות עבור מוצר והעשוי לכלול דרישות תפקודיות, דרישות ביצועים, דרישות מִנשָׁק, דרישות תכן, דרישות סביבתיות ותקני פיתוח.

20.07.14 **תִּיק פְּרוֹיֶקְט** **project file**
**project notebook**

מאגר מרכזי של חומר המתייחס לפרויקט, הכולל בדרך כלל תזכירים, תוכניות, דוחות טכניים ופריטים קשורים.

20.07.15 **תָּכְנִית פְּרוֹיֶקְט** **project plan**

מסמך המתאר את הגישה הטכנית ואת גישת הניהול של הפרויקט, הכולל בדרך כלל תיאור של העבודה, של המשאבים, של השיטות, של הנהלים, של לוחות הזמנים ושל צורת הארגון של הפרויקט.

20.07.16 **תִּיק פִּתּוּחַ תָּכְנָה** **software development file, SDF** [abb]
**תִּיק פִּיתּוּחַ יְחִידָה** **software development folder**
**software development notebook**
**unit development folder, UDF** [abb]

אוסף החומרים המתייחסים לפיתוח של יחידת תוכנה נתונה או של קבוצת יחידות קשורות.

20.07.17 **תָּכְנִית בְּדִיקָה** **test plan**
**תָּכְנִית בְּדִיקָה וְהַעֲרָכָה שֶׁל מַעֲרֶכֶת** **system test and evaluation plan**

מסמך הקובע דרישות מפורטות, קריטריונים, מתודולוגיה כללית, אחריות ותכנון כללי לצורך בדיקה והערכה של מערכת.

20.07.18 **טַבְלַת הַחְלָטָה** **decision table**

טבלה של כל האפשרויות בתיאור בעיה ושל הפעולה שתינקט בכל אחד מהמצבים.

20.07.19 **דּוּחַ הַעֲרָכָה** **evaluation report**

דוח מעקב מערכת, המתאר עבור כל שלב פיתוח איך ובאיזו מידה הושגו יעדי המערכת באותו שלב, המזהה את הבעיות הנותרות והמיועד לעזור בהמשך הפיתוח.

20.07.20 **הוֹדָעַת שִׁנּוּי מִפְרָט** **specification change notice**
**SCN** [abb]

בניהול תצורה - מסמך המשמש להצעה, להעברה ולרישום של שינויים במפרט.

20.07.21 **דּוּחַ אֵרוּעַ בְּדִיקָה** **test incident report**

מסמך המתאר אירוע שהתרחש במהלך בדיקה והדורש חקירה נוספת.

20.07.22 **מִסְמַךְ תֵּאוּר גִּרְסָה** **version description document**
**VDD** [abb]

מסמך המלווה והמזהה גרסה נתונה של מערכת או של רכיב, הכולל בדרך כלל רשימת מצאי של חלקי מערכת או של חלקי רכיב, זיהוי שינויים ששולבו בגרסה זו ומידע על ההתקנה והתפעול הייחודיים לה.

20.07.23 **תַּרְשִׁים-רֶשֶׁת שֶׁל תָּכְנִיּוֹת** **program network chart**

תרשים המציג את היוחסה בין כמה תוכניות מחשב.

20.07.24 **מַדְרִיךְ לַתָּכְנִיתָן** **programmer manual**

מסמך המספק מידע הנדרש לפיתוח ולהסגלה (modification) של תוכנה של מערכת מחשב נתונה, הכולל בדרך כלל תיאור של תצורת ציוד, של אופייני תפעול, של מאפייני תכנות, של מאפייני קלט/פלט ושל מאפייני הידור או איסוף במערכת המחשב.

20.07.25 **מַדְרִיךְ תַּחְזוּקַת תָּכְנִית** **program maintenance manual**

מסמך המספק את המידע הנדרש כדי לתחזק תוכנית.

20.07.26 **מַדְרִיךְ אִבְחוּן** **diagnostic manual**

מסמך המספק את המידע הנדרש לביצוע נוהלי אבחון של מערכת או של רכיב במטרה לזהות ליקויי תפקוד ולתקנם.

20.07.27 **מַדְרִיךְ הַתְקָנָה** **installation manual**

מסמך המספק את המידע הנדרש כדי להתקין מערכת או רכיב, לקבוע פרמטרים תחיליים ולהכין את המערכת או את הרכיב לשימוש תפעולי.

20.07.28 **מַדְרִיךְ תַּחְזוּקָה** **maintenance manual**

מסמך המספק את המידע הנדרש לתחזוק יחידה תפקודית.

20.07.29 **מַדְרִיךְ תְּמִיכָה** **support manual**

מסמך המספק את המידע הנדרש לשירות ולתחזוק של מערכת או של רכיב תפעוליים לכל אורך מחזור החיים.

**הערה**:

המסמך כולל בדרך כלל תיאור של החומרה ושל התוכנה המרכיבות את המערכת או את הרכיב, ונהלים לשירות, לתיקון או לתכנות מחדש.

20.07.30 **מַדְרִיךְ לַמַּפְעִיל** **operator manual**

מסמך המספק את המידע הנדרש כדי לתחל ולתפעל מערכת או רכיב, הכולל בדרך כלל תיאור הנהלים להכנה, לתפעול, לניטור ולהתאוששות.

**הערה**:

מבחינים בין 'מדריך למפעיל' לבין 'מדריך למשתמש' כאשר ניתן להבחין בין מי שמפעילים מערכת מחשב (מרכיבים סרטים מגנטיים וכדומה) לבין מי שמשתמשים במערכת למטרותיה המיועדות.

20.07.31 **מַדְרִיךְ לַמִּשְׁתַּמֵּשׁ** **user manual**

**user's guide**

מסמך המספק את המידע הנדרש להפעלת מערכת או רכיב כדי להשיג תוצאות רצויות, הכולל בדרך כלל תיאור יכולות של המערכת או של הרכיב, מגבלות, אפשרויות, קלט מותר, פלט צפוי, הודעות שגיאה אפשריות והוראות מיוחדות.

**הערה**:

מבחינים בין 'מדריך למפעיל' לבין 'מדריך למשתמש' כאשר ניתן להבחין בין מי שמפעילים מערכת מחשב (מרכיבים סרטים מגנטיים וכדומה) לבין מי שמשתמשים במערכת למטרותיה המיועדות.

## 20.08 - ניהול מיזמים ובקרתם - Project Management and Control

20.08.01 **פְּרוֹיֶקְט** **Project**

משימה שהיעדים, ההיקף ומשך הזמן שלה מוגדרים מראש.

20.08.02 **נִהוּל פְּרוֹיֶקְט** **project management**

הפעילויות הקשורות לתכנון פרויקט ולבקרת פרויקט.

20.08.03 **תִּכְנוּן פְּרוֹיֶקְט** **project planning**

הפעילויות הקשורות להגדרת הרכיבים, התזמון, המשאבים, הנהלים והתהליכים של הפרויקט.

20.08.04 **מִפְרַט פְּרוֹיֶקְט** **project specification**

מִפְרט היעדים, הדרישות וההיקף של פרויקט, וקשריו לפרויקטים אחרים.

20.08.05 **בַּקָּרַת פְּרוֹיֶקְט** **project control**

הפעילויות הקשורות לניטור ההתקדמות של פרויקט, איכותו וניצול משאביו, בהשוואה לתוכניות הפרויקט.

20.08.06 **תַּרְשִׁים רֶשֶׁת** **network chart**

בבקרת פרויקט - גרף מכוון **(directed graph)** המשמש לתיאור ולזִמנון **(scheduling)** של אירועים, של פעילויות ושל הַקשרים ביניהם.

20.08.07 **תִּכְנוּן רֶשֶׁת** **network planning**

טכניקה המשתמשת בתרשימי רשת לשם תכנון, זִמנון **(scheduling)** ובקרה של פרויקט.

20.08.08 **בַּקָּרַת תְּצוּרָה** **configuration control**

**בַּקָּרַת שִׁנּוּיִים** **change control**

אלמנט בניהול תצורה המורכב מהערכה, מתיאום, מאישור או מדחייה, ומימוש של שינויים בפריטי תצורה לאחר הקביעה הפורמלית של מזהי התצורה שלהם.

20.08.09 **וַעֲדָה לְבַקָּרַת תְּצוּרָה** **configuration control board**

**CCB** [abb]

**change control board**

צוות בר-סמכא המעריך הצעות לשינויים בקו הבסיס לפיתוח הנוכחי של מוצר, ומאשר או פוסל אותן.

20.08.10 **תָּכְנִיתָן רָאשִׁי** **chief programmer**

**רא"ש צֶוֶת-תָּכְנִיתָן רָאשִׁי**

תוכניתן בכיר שהוא בעל שליטה מלאה, שתחום האחריות שלו כולל: הפקת חלקי מפתח של התוכנה שהוקצו לצוות, תיאום עבודת הצוות ובדיקת העבודה של חברים אחרים בצוות, ושיש לו הבנה טכנית כוללת של התוכנה המפותחת ואחריות עליה.

20.08.11 **צֶוֶת-תָּכְנִיתָן רָאשִׁי** **chief programmer team**

קבוצת פיתוח תוכנה המורכבת מתוכניתן ראשי, תוכינתן גיבוי, מזכיר או ספרן ותוכניתנים ומומחים נוספים כנדרש, והמפעילה נהלים שתוכננו לעשות שימוש מיטבי בכישורים של כל חבר בצוות.

20.08.12 **שִׁינוּי הַנְדָּסִי** **engineering change**

בניהול תצורה - שינוי בתצורה של פריט-תצורה או של פריט מוגדר אחר, הנעשה אחרי הקביעה הפורמלית של מזהה התצורה שלו.

20.08.13 **הַצָעַת שִׁינוּי הַנְדָסִי** **engineering change proposal**

**ECP** [abb]

בניהול תצורה - שינוי הנדסי מוצע והתיעוד המתאר אותו.

20.08.14 **כְּתַב וִתּוּר** **Waiver**

הרשאה בכתב לקבל פריט תצורה או פריט מוגדר אחר, שבמהלך ייצורו או לאחר שהוגש לביקורת נמצא שהוא חורג מהדרישות שפורטו, ואף על פי כן נחשב מתאים לשימוש במצבו הנוכחי או לאחר עיבוד מחדש בשיטה מאושרת.

# רשימת מונחים עברית-אנגלית

| מספר מונח | מונח עברי | English Term |
|---|---|---|
| 20.03.58 | אֲרִיזָה | packaging |
| 20.03.26 | אִחוּד מוֹדוּלִים | demodularization |
| 20.05.49 | אִמּוּת | verification |
| 20.05.01 | אֵיכוּת | quality |
| 20.05.53 | אֵרוּעַ בדיקת תֹּכְנָה | Software test incident |
| 20.01.28 | אַב-טִיפּוּס | prototype |
| 20.02.29 | אַמְדָן | estimator |
| 20.05.54 | אַרְכִיטֶקְטוּרַת עֶרְכַּת בְּדִיקָה | test set architecture |
| 20.01.11 | בְּעָיַת יִשּׁוּם | application problem |
| 20.05.25 | בְּדִיקַת אַלְפָא | alpha testing |
| 20.05.38 | בְּדִיקַת בִּצּוּעִים | performance testing |
| 20.05.26 | בְּדִיקַת בֵּיטָא | beta testing |
| 20.05.33 | בְּדִיקַת גַּב-אֶל-גַּב | back-to-back testing |
| 20.05.45 | בְּדִיקַת יְחִידָה | unit testing |
| 20.05.51 | בְּדִיקַת יחידה | unit test |
| 20.05.39 | בְּדִיקַת כְּשֵׁירוּת | qualification testing |
| 20.05.31 | בְּדִיקַת תְּכְלוּל | integration test |
| 20.05.34 | בְּדִיקַת לוּלָאָה | loopback testing |
| 20.05.43 | בְּדִיקַת מִבְנֶה | structural testing |
| 20.05.32 | בְּדִיקַת מִנְשָׁק | interface testing |
| 20.05.41 | בְּדִיקַת מִשְׁפָּטִים | statement testing |
| 20.05.42 | בְּדִיקַת מַאֲמָץ | stress testing |
| 20.05.44 | בְּדִיקַת מַעֲרֶכֶת | system testing |
| 20.05.35 | בְּדִיקַת מוּטַצְיָה | mutation testing |
| 20.05.37 | בְּדִיקַת נָתִיב | path testing |
| 20.05.28 | בְּדִיקַת פִּיתּוּחַ | developmental testing |
| 20.05.47 | בְּדִיקַת קַבָּלָה | acceptance test |
| 20.05.52 | בְּדִיקַת שְׁמִישׁוּת | usability test |
| 20.05.40 | בְּדִיקַת תְּסוּגָה | regression testing |
| 20.05.36 | בְּדִיקַת תִּפְעוּל | operation testing |
| 20.05.30 | בְּדִיקָה לא-פוֹרְמָלִית | informal testing |
| 20.05.29 | בְּדִיקָה פוֹרְמָלִית | formal testing |
| 20.05.27 | בְּדִיקָה תִּפְקוּדִית | functional testing |
| 20.05.46 | בִּדּוּק יְדָנִי | desk checking |
| 20.06.07 | בִּדּוּק שִׁחְרוּר | checkout |
| 20.02.21 | בִּטוּי נָתִיב | path expression |
| 20.03.55 | בִּיקֹרֶת נוֹקֶבֶת | walk-through |
| 20.03.64 | בַּסֵּס מֵחָדָשׁ | rebase (v) |
| 20.05.61 | בִּצּוּעַ חוֹזֵר | reversible execution |
| 20.05.24 | בְּדִיקָה | testing |
| 20.01.52 | בְּדִיקָתִיּוּת | testability |

| English Term | מונח עברי | מספר מונח |
|---|---|---|
| interface control | בַּקָּרַת מִנְשָׁק | 20.03.38 |
| configuration control | בַּקָּרַת תְּצוּרָה | 20.08.08 |
| quality control | בַּקָּרַת אֵיכוּת | 20.05.03 |
| project control | בַּקָּרַת פְּרוֹיֶקְט | 20.08.05 |
| benchmark test | בּוֹחַן עַל=פִּי נְקֻדַּת מַדּוּד | 20.05.48 |
| code inspection | בְּחִינַת קוֹד | 20.05.18 |
| design inspection | בְּחִינַת תֶּכֶן | 20.05.19 |
| version | גִּרְסָה | 20.01.37 |
| interface requirement | דְּרִישַׁת מִנְשָׁק | 20.02.10 |
| performance requirement | דְּרִישַׁת בִּצּוּעִים | 20.02.11 |
| implementation requirement | דְּרִישַׁת מִמּוּשׁ | 20.02.09 |
| requirement | דְּרִישָׁה | 20.01.05 |
| downward compression | דְּחִיסָה מַטָּה | 20.03.27 |
| upward compression | דְּחִיסָה מַעְלָה | 20.03.29 |
| lateral compression | דְּחִיסָה צִדִּית | 20.03.28 |
| environmental requirement | דְּרִישָׁה סְבִיבָתִית | 20.02.07 |
| physical requirement | דְּרִישָׁה פִּיזִיקָלִית | 20.02.12 |
| functional requirement | דְּרִישָׁה תִּפְקוּדִית | 20.02.08 |
| configuration status accounting | דִּווּחַ מִצַּב הַתְּצוּרָה | 20.03.39 |
| thrashing | דִּשְׁדּוּשׁ | 20.06.12 |
| coexistence | דּוּ-קִיּוּם | 20.04.12 |
| test incident report | דּוּחַ אֵרוּעַ בְּדִיקָה | 20.07.21 |
| evaluation report | דּוּחַ הַעֲרָכָה | 20.07.19 |
| structure clash | הִתְנַגְּשׁוּת מִבְנִים | 20.03.45 |
| order clash | הִתְנַגְּשׁוּת סְדָרִים | 20.03.44 |
| cutover | הֶחְלֵף | 20.04.04 |
| test driver | הֶנֵּעַ בְּדִיקָה | 20.05.56 |
| abstraction (1) | הֶפְשֵׁט [ש"ע] | 20.02.04 |
| quality assurance | הַבְטָחַת אֵיכוּת | 20.05.02 |
| problem definition | הַגְדָּרַת בְּעָיָה | 20.02.03 |
| simulation | הַדְמָיָה | 20.01.32 |
| prototyping | הַטְפָּסַת-אָב | 20.01.29 |
| rapid prototyping | הַטְפָּסַת-אָב מְהִירָה | 20.03.18 |
| computer-aided software engineering | הַנְדָּסַת תֹּכְנָה בְּעֶזְרַת מַחְשֵׁב | 20.04.01 |
| computer performance evaluation | הַעֲרָכַת בִּצּוּעֵי מַחְשֵׁב | 20.05.15 |
| data abstraction | הַפְשָׁטַת נְתוּנִים | 20.03.14 |
| abstraction (2) | הַפְשָׁטָה [שם פעולה] | 20.02.05 |
| pipelining | הַצְנָרָה | 20.03.57 |
| engineering change proposal | הַצָּעַת שִׁינּוּי הַנְדָּסִי | 20.08.13 |
| certification | הַתְעָדָה | 20.05.11 |
| specification change notice | הוֹדָעַת שִׁנּוּי מִפְרָט | 20.07.20 |
| parent | הוֹרֶה | 20.01.66 |
| configuration control board | וַעֲדָה לְבַקָּרַת תְּצוּרָה | 20.08.09 |
| configuration identification | זִהוּי תְּצוּרָה | 20.03.35 |
| software package | חֲבִילַת תֹּכְנָה | 20.01.61 |

| English Term | מונח עברי | מספר מונח |
|---|---|---|
| critical piece first | חֵלֶק קְרִיטִי תְּחִילָה | 20.03.15 |
| opportunity study | חֵקֶר הִזְדַּמְּנוּת | 20.02.01 |
| feasibility study | חֵקֶר הִתָּכְנוּת | 20.02.02 |
| robustness | חֹסֶן | 20.01.47 |
| decision table | טַבְלַת הַחְלָטָה | 20.07.18 |
| user portability | יְבִילוּת מִשְׁתַּמֵּשׁ | 20.01.57 |
| design unit | יְחִידַת תָּכָן | 20.03.09 |
| application | יִשּׁוּם | 20.01.10 |
| redundancy | יַתִּירוּת | 20.06.08 |
| homogeneous redundance | יַתִּירוּת אֲחִידָה | 20.06.11 |
| standby redundancy | יַתִּירוּת בַּעֲתוּדָה | 20.06.10 |
| active redundancy | יַתִּירוּת פָּעִילָה | 20.06.09 |
| system integrity | כְּלִילוּת מַעֲרֶכֶת | 20.05.10 |
| qualification | כְּשִׁירוּת | 20.05.12 |
| waiver | כְּתַב וִתּוּר | 20.08.14 |
| system integration | תִּכְלוּל מַעֲרָכוֹת | 20.04.02 |
| extendability | כֹּשֶׁר הִתְרַחֲבוּת | 20.01.43 |
| test generator | מְחוֹלֵל בְּדִיקוֹת | 20.05.55 |
| data inventory | מְצַאי נְתוּנִים | 20.01.26 |
| self-descriptive | מְתֹאָר מֵאֵלָיו | 20.01.48 |
| self-documented | מְתֹעָד מֵאֵלָיו | 20.01.49 |
| physical configuration audit | מִבְדַּק תְּצוּרָה פִּיזִי | 20.05.21 |
| functional configuration audit | מִבְדַּק תְּצוּרָה תִּפְקוּדִי | 20.05.20 |
| instrumentation | מִכְשׁוּר | 20.06.13 |
| user interface | מִנְשָׁק מִשְׁתַּמֵּשׁ | 20.01.56 |
| requirements document | מִסְמַךְ דְּרִישׁוֹת | 20.07.13 |
| version description document | מִסְמַךְ תֵּאוּר גִּרְסָה | 20.07.22 |
| design document | מִסְמַךְ תָּכָן | 20.07.04 |
| software requirements specification | מִפְרַט דְּרִישׁוֹת תָּכְנָה | 20.07.05 |
| test case specification | מִפְרַט מִקְרֵה בְּדִיקָה | 20.07.08 |
| project specification | מִפְרַט פְּרוֹיֶקְט | 20.08.04 |
| program specification | מִפְרַט תָּכְנִית | 20.07.06 |
| functional specification | מִפְרַט תִּפְקוּדִי | 20.07.07 |
| specification | מִפְרָט | 20.01.06 |
| product specification | מִפְרָט מוּצָר | 20.01.08 |
| formal specification | מִפְרָט פּוֹרְמָלִי | 20.01.07 |
| quality metric | מֶטְרִיקַת אֵיכוּת | 20.05.05 |
| software quality metric | מֶטְרִיקַת אֵיכוּת | 20.05.65 |
| metric | מֶטְרִיקָה | 20.05.06 |
| diagnostic manual | מַדְרִיךְ אִבְחוּן | 20.07.26 |
| installation manual | מַדְרִיךְ הַתְקָנָה | 20.07.27 |
| operator manual | מַדְרִיךְ לַמַּפְעִיל | 20.07.30 |
| user manual | מַדְרִיךְ לַמִּשְׁתַּמֵּשׁ | 20.07.31 |
| programmer manual | מַדְרִיךְ לַתְּכְנִיתָן | 20.07.24 |
| support manual | מַדְרִיךְ תְּמִיכָה | 20.07.29 |

| English Term | מונח עברי | מספר מונח |
|---|---|---|
| program maintenance manual | מַדְרִיךְ תַּחְזוּקַת תָּכְנִית | 20.07.25 |
| maintenance manual | מַדְרִיךְ תַּחְזוּקָה | 20.07.28 |
| software lifecycle | מַחֲזוֹר חַיֵּי תָּכְנָה | 20.01.62 |
| system lifecycle | מַחֲזוֹר חַיֵּי מַעֲרֶכֶת | 20.01.21 |
| software development cycle | מַחֲזוֹר פִּתּוּחַ תָּכְנָה | 20.01.04 |
| traceability matrix | מַטְרִיצַת עֲקִיבוּת | 20.03.42 |
| transaction matrix | מַטְרִיצַת תְּנוּעוֹת | 20.03.43 |
| bottom-up | מַטָּה-מַעְלָה | 20.01.17 |
| top-down | מַעְלָה-מַטָּה | 20.01.16 |
| system follow-up | מַעֲקַב מַעֲרֶכֶת | 20.02.26 |
| menu by-pass | מַעֲקַף תַּפְרִיט | 20.03.48 |
| automated verification system | מַעֲרֶכֶת אִמּוּת מְמֻחְשֶׁבֶת | 20.05.57 |
| change-over system | מַעֲרֶכֶת מַעֲבָר | 20.04.03 |
| closed system | מַעֲרֶכֶת סְגוּרָה | 20.01.18 |
| comparator | מַשְׁוָן | 20.05.58 |
| deliver (v) | מָסַר | 20.01.68 |
| command-driven | מֻנְחֵה פְּקֻדּוֹת | 20.03.46 |
| menu-driven | מֻנְחֵה תַּפְרִיטִים | 20.03.47 |
| implementation of a system | מִמּוּשׁ מַעֲרֶכֶת | 20.01.35 |
| entity-relationship model | מוֹדֵל יֵשֵׁיּוֹת-יַחְסוֹת | 20.02.27 |
| Open Information Model | מוֹדֵל מֵידַע פָּתוּחַ | 20.02.28 |
| waterfall model | מוֹדֵל מַפַּל-מַיִם | 20.03.22 |
| spiral model | מוֹדֵל סְפִּירָלִי | 20.03.21 |
| workload model | מוֹדֵל עֹמֶס עֲבוֹדָה | 20.05.17 |
| program mutation | מוּטַצְיָה שֶׁל תָּכְנִית | 20.05.60 |
| software product | מוּצַר תָּכְנָה | 20.01.14 |
| efficiency | נְצִילוּת | 20.01.41 |
| execution efficiency | נְצִילוּת בְּצוּעַ | 20.01.42 |
| benchmark | נְקֻדַּת מִדּוּד | 20.05.66 |
| system data | נְתוּנֵי מַעֲרֶכֶת | 20.04.11 |
| project management | נִהוּל פְּרוֹיֶקְט | 20.08.02 |
| configuration management | נִהוּל תְּצוּרָה | 20.03.36 |
| utilization | נִיצוּל | 20.01.58 |
| deliverable | נִמְסָר | 20.01.69 |
| requirements analysis | נִתּוּחַ דְּרִישׁוֹת | 20.02.06 |
| dynamic analysis | נִתּוּחַ דִּינָמִי | 20.05.22 |
| transform analysis | נִתּוּחַ הַתְּמָרָה | 20.03.25 |
| information analysis | נִתּוּחַ מֵידָע | 20.02.17 |
| system analysis | נִתּוּחַ מַעֲרָכוֹת | 20.02.15 |
| path analysis | נִתּוּחַ נְתִיבִים | 20.02.19 |
| data analysis | נִתּוּחַ נְתוּנִים | 20.02.18 |
| transaction analysis | נִתּוּחַ תְּנוּעוֹת | 20.03.24 |
| functional analysis | נִתּוּחַ תִּפְקּוּדִי | 20.02.16 |
| static analysis | נִתּוּח סְטָטִי | 20.05.23 |
| timing analyzer | נַתֵּחַ תִּזְמוּן | 20.01.54 |

| English Term | מונח עברי | מספר מונח |
|---|---|---|
| variant | נֹסַח אַחֵר | 20.05.59 |
| deviation | סְטִיָה | 20.01.40 |
| software development library | סִפְרִיַת פִּתּוּחַ תָּכְנָה | 20.04.10 |
| reference book | סֵפֶר | 20.01.63 |
| fault-rate threshold | סַף קֶצֶב תַּקָּלוֹת | 20.05.64 |
| fault threshold | סַף תַּקָּלוֹת | 20.05.63 |
| traceability | עֲקִיבוּת | 20.05.07 |
| stepwise refinement | עִדּוּן צַעַד-צַעַד | 20.03.17 |
| fault trace | עֵקֶבֶת תַּקָּלוֹת | 20.05.62 |
| workload | עֹמֶס עֲבוֹדָה | 20.05.16 |
| system development | פִּתּוּחַ מַעֲרָכוֹת | 20.01.01 |
| critical item | פְּרִיט קָרִיטִי | 20.03.13 |
| configuration item | פְּרִיט תְּצוּרָה | 20.03.10 |
| hardware configuration item | פְּרִיט תְּצוּרַת חֲומָרָה | 20.03.12 |
| computer software configuration item | פְּרִיט תְּצוּרַת תָּכְנָה | 20.03.11 |
| project | פְּרוֹיֶקְט | 20.08.01 |
| pilot project | פְּרוֹיֶקְט חָלוּץ | 20.01.30 |
| system profile | פְּרוֹפִיל מַעֲרֶכֶת | 20.05.09 |
| component-based development | פִּתּוּחַ מְבוּסַס רְכִיבִים | 20.03.62 |
| incremental development | פִּתּוּחַ תּוֹסַפְתִּי | 20.03.16 |
| factoring | פֵּרוּק לְגוֹרְמִים | 20.03.33 |
| modular decomposition | פֵּרוּק לְמוֹדוּלִים | 20.03.30 |
| hierarchical decomposition | פֵּרוּק מַדְרָגִי | 20.03.32 |
| functional decomposition | פֵּרוּק תִּפְקוּדִי | 20.03.31 |
| pipe | צִנּוֹר | 20.03.56 |
| descendant | צֶאֱצָא | 20.01.67 |
| chief programmer team | צֶוֶת-תָּכְנִיתָן רָאשִׁי | 20.08.11 |
| pipeline | צַנֶּרֶת | 20.03.20 |
| acceptance criterion | קְרִיטֶרְיוֹן קַבָּלָה | 20.05.08 |
| software diversity | קִדּוּד מִתְחָרֶה | 20.04.08 |
| input-process-output | קֶלֶט-עִבּוּד-פֶּלֶט | 20.03.19 |
| product baseline | קַו-בָּסִיס לְמוּצָר | 20.07.12 |
| developmental baseline | קַו-בָּסִיס לְפִתּוּחַ | 20.07.10 |
| allocated baseline | קַו-בָּסִיס מֻקְצֶה | 20.07.09 |
| functional baseline | קַו-בָּסִיס תִּפְקוּדִי | 20.07.11 |
| baseline | קַו בָּסִיס | 20.01.09 |
| black box | קֻפְסָה שְׁחוֹרָה | 20.03.40 |
| glass box | קֻפְסָה שְׁקוּפָה | 20.03.41 |
| field-level sensitivity | רְגִישׁוּת בְּרָמַת הַשָּׂדֶה | 20.03.61 |
| parallel run | רִיצָה בְּמַקְבִּיל | 20.04.05 |
| petri net | רֶשֶׁת פֶּטְרִי | 20.02.25 |
| reusable | רַב-שִׁמּוּשִׁי | 20.01.46 |
| reusability | רַב-שִׁמּוּשִׁיּוּת | 20.01.45 |
| system development phase | שְׁלַב פִּתּוּחַ מַעֲרֶכֶת | 20.01.19 |
| reuse | שִׁמּוּשׁ חוֹזֵר | 20.01.64 |

| English Term | מונח עברי | מספר מונח |
|---|---|---|
| lifecycle phase | שְׁלַב בְּמַחֲזוֹר חַיִּים | 20.01.20 |
| installation and checkout phase | שְׁלַב (הַ)הַתְקָנָה וּבְדִּיק (הַ)שִׁחְרוּר | 20.01.33 |
| test phase | שְׁלַב הַבְּדִיקָה | 20.01.31 |
| requirements phase | שְׁלַב הַדְּרִישׁוֹת | 20.01.22 |
| concept phase | שְׁלַב הַהַמְשְׂגָה | 20.01.23 |
| retirement phase | שְׁלַב הַזְּנִיחָה | 20.01.39 |
| implementation phase | שְׁלַב הַמִּמּוּשׁ | 20.01.34 |
| operation and maintenance phase | שְׁלַב הַתִּפְעוּל וְהַתַּחְזוּקָה | 20.01.36 |
| design phase | שְׁלַב הַתָּכָן | 20.01.24 |
| right-sizing | שִׁדּוּד הוֹלֵם | 20.06.18 |
| resizing | שִׁדּוּד מֵחָדָשׁ | 20.06.17 |
| downsizing | שִׁדּוּד מַטָּה | 20.06.16 |
| upsizing | שִׁדּוּד מַעְלָה | 20.06.15 |
| system sizing | שִׁדּוּד מַעְרָכוֹת | 20.06.14 |
| engineering change | שִׁינּוּי הַנְדָּסִי | 20.08.12 |
| usability | שִׁמּוּשִׁיּוּת | 20.01.55 |
| serviceability | שֵׁרוּתִיּוּת | 20.01.50 |
| sizing | שִׁדּוּד | 20.01.51 |
| software development process | תַּהֲלִיךְ פִּתּוּחַ תָּכְנָה | 20.01.02 |
| software requirements review | תַּסְקִיר דְּרִישׁוֹת תָּכְנָה | 20.02.14 |
| test plan | תָּכְנִית בְּדִיקָה | 20.07.17 |
| application software | תָּכְנִית יִשּׁוּם | 20.01.13 |
| application software | תָּכְנַת יִשּׁוּמִים | 20.01.12 |
| support software | תָּכְנַת תְּמִיכָה | 20.01.15 |
| quality attribute | תְּכוּנַת אֵיכוּת | 20.05.04 |
| system support | תְּמִיכָה בַּמַּעֲרֶכֶת | 20.01.38 |
| path condition | תְּנָאֵי נָתִיב | 20.02.20 |
| transaction | תְּנוּעָה | 20.03.23 |
| developmental configuration | תְּצוּרַת פִּתּוּחַ | 20.04.06 |
| form, fit and function | תְּצוּרָה כְּלָלִית | 20.03.37 |
| simulated configuration | תְּצוּרָה מֻדְמֵית | 20.03.63 |
| timing | תִּזְמוּן | 20.01.53 |
| project file | תִּיק פְּרוֹיֶקְט | 20.07.14 |
| software development file, SDF [abb] | תִּיק פִּתּוּחַ תָּכְנָה | 20.07.16 |
| integrate | תִּכְלֵל | 20.03.59 |
| project planning | תִּכְנוּן פְּרוֹיֶקְט | 20.08.03 |
| network planning | תִּכְנוּן רֶשֶׁת | 20.08.07 |
| egoless programming | תִּכְנוּת לאֹ-אָנוֹכִי | 20.04.07 |
| user documentation | תִּעוּד מִשְׁתַּמֵּשׁ | 20.07.03 |
| system documentation | תִּעוּד מַעֲרֶכֶת | 20.07.01 |
| validation | תִּקּוּף | 20.05.50 |
| system description | תֵּאוּר מַעֲרֶכֶת | 20.07.02 |
| software design description | תֵּאוּר תָּכָן תָּכְנָה | 20.03.08 |
| architectural design | תָּכָן אַרְכִּיטֶקְטוֹנִי [הפעולה] | 20.03.02 |
| architectural design | תָּכָן אַרְכִּיטֶקְטוֹנִי [התוצאה] | 20.03.03 |

| English Term | מונח עברי | מספר מונח |
|---|---|---|
| object-oriented design | תֶּכֶן כִּוּוּן-עֲצָמִים | 20.01.65 |
| object-oriented design | תֶּכֶן כִּוּוּן עֲצָמִים | 20.03.07 |
| detailed design | תֶּכֶן מְפוֹרָט | 20.01.27 |
| system design | תֶּכֶן מַעֲרָכוֹת | 20.03.01 |
| conceptual system design | תֶּכֶן מַעֲרָכוֹת מֻושְׁגִי | 20.03.04 |
| structured design | תֶּכֶן מֻבְנֶה | 20.03.06 |
| preliminary design | תֶּכֶן רֵאשׁוֹנִי | 20.01.25 |
| functional design | תֶּכֶן תִּפְקּוּדִי | 20.03.05 |
| system maintenance | תַּחְזוּקַת מַעֲרָכֶת | 20.06.02 |
| maintenance | תַּחְזוּקָה | 20.06.01 |
| adaptive maintenance | תַּחְזוּקָה מְסַגֶּלֶת | 20.06.03 |
| perfective maintenance | תַּחְזוּקָה מְשַׁכֶּלֶת | 20.06.05 |
| corrective maintenance | תַּחְזוּקָה מְתַקֶּנֶת | 20.06.04 |
| preventive maintenance | תַּחְזוּקָה מוֹנַעַת | 20.06.06 |
| maintainability | תַּחְזוּקָתִיּוּת | 20.01.44 |
| requirements review | תַּסְקִיר דְּרִישׁוֹת | 20.02.13 |
| formal qualification review | תַּסְקִיר כְּשִׁירוּת פּוֹרְמָלִי | 20.05.13 |
| test readiness review | תַּסְקִיר מוּכָנוּת לִבְדִיקָה | 20.05.14 |
| code review | תַּסְקִיר קוֹד | 20.04.09 |
| design review | תַּסְקִיר תֶּכֶן | 20.03.51 |
| system design review | תַּסְקִיר תֶּכֶן-מַעֲרָכֶת | 20.03.53 |
| preliminary design review | תַּסְקִיר תֶּכֶן-קַדְמִי | 20.03.52 |
| critical design review | תַּסְקִיר תֶּכֶן-קְרִיטִי | 20.03.54 |
| entity-relationship diagram, E-R diagram | תַּרְשִׁים יֵשׁוּיוֹת-יַחְסוֹת | 20.02.24 |
| data-structure-centered diagram | תַּרְשִׁים מְמֻרְכָּז מִבְנֵה-נְתוּנִים | 20.02.22 |
| data-structure diagram | תַּרְשִׁים מִבְנֵה נְתוּנִים | 20.02.23 |
| program network chart | תַּרְשִׁים-רֶשֶׁת שֶׁל תָּכְנִיּוֹת | 20.07.23 |
| network chart | תַּרְשִׁים רֶשֶׁת | 20.08.06 |
| configuration diagram | תַּרְשִׁים תְּצוּרָה | 20.03.34 |
| integrated infrastructure | תַּשְׁתִּית מְתֻכְלֶלֶת | 20.03.60 |
| chief programmer | תָּכְנִיתָן רָאשִׁי | 20.08.10 |
| project plan | תָּכְנִית פְּרוֹיֶקְט | 20.07.15 |
| critical software | תָּכְנָה קְרִיטִית | 20.01.60 |
| system software | תָּכְנַת מַעֲרָכֶת | 20.01.59 |
| downward compatible | תּוֹאֵם מַטָּה | 20.03.49 |
| upward compatible | תּוֹאֵם מַעְלָה | 20.03.50 |
| software development plan | תוכנית-פרויקט לפרויקט פיתוח תוכנה. | 20.01.03 |

# רשימת מונחים אנגלית-עברית

| English Term | מונח עברי | מספר מונח |
|---|---|---|
| abstraction (1) | הֶפְשֵׁט [ש"ע] | 20.02.04 |
| abstraction (2) | הַפְשָׁטָה [שם פעולה] | 20.02.05 |
| acceptance criterion | קְרִיטֶרְיוֹן קַבָּלָה | 20.05.08 |
| acceptance test | בְּדִיקַת קַבָּלָה | 20.05.47 |
| active redundancy | יַתִּירוּת פְּעִילָה | 20.06.09 |
| adaptive maintenance | תַּחְזוּקָה מְסַגֶּלֶת | 20.06.03 |
| allocated baseline | קַו-בָּסִיס מֻקְצֶה | 20.07.09 |
| alpha testing | בְּדִיקַת אַלְפָא | 20.05.25 |
| application | יִשּׁוּם | 20.01.10 |
| application problem | בְּעָיַת יִשּׁוּם | 20.01.11 |
| application software | תָּכְנַת יִשּׁוּמִים | 20.01.12 |
| application software | תָּכְנִית יִשּׁוּם | 20.01.13 |
| architectural design | תֶּכֶן אַרְכִּיטֶקְטוֹנִי [הפעולה] | 20.03.02 |
| architectural design | תֶּכֶן אַרְכִּיטֶקְטוֹנִי [התוצאה] | 20.03.03 |
| automated verification system | מַעֲרֶכֶת אִמּוּת מְמֻחְשֶׁבֶת | 20.05.57 |
| back-to-back testing | בְּדִיקַת גַּב-אֶל-גַּב | 20.05.33 |
| baseline | קַו בָּסִיס | 20.01.09 |
| benchmark | נְקֻדַּת מִדּוּד | 20.05.66 |
| benchmark test | בּוֹחַן עַל=פִּי נְקֻדַּת מִדּוּד | 20.05.48 |
| beta testing | בְּדִיקַת בֵּיתָא | 20.05.26 |
| black box | קֻפְסָה שְׁחוֹרָה | 20.03.40 |
| bottom-up | מַטָּה-מַעְלָה | 20.01.17 |
| certification | הַתְעָדָה | 20.05.11 |
| change-over system | מַעֲרֶכֶת מַעֲבָר | 20.04.03 |
| checkout | בְּדּוּק שִׁחְרוּר | 20.06.07 |
| chief programmer | תָּכְנִיתָן רָאשִׁי | 20.08.10 |
| chief programmer team | צֶוֶת-תָּכְנִיתָן רָאשִׁי | 20.08.11 |
| closed system | מַעֲרֶכֶת סְגוּרָה | 20.01.18 |
| code inspection | בְּחִינַת קוֹד | 20.05.18 |
| code review | תַּסְקִיר קוֹד | 20.04.09 |
| coexistence | דּוּ-קִיּוּם | 20.04.12 |
| command-driven | מֻנְחֵה פְּקֻדּוֹת | 20.03.46 |
| comparator | מַשְׁוָן | 20.05.58 |
| component-based development | פִּתּוּחַ מְבוּסַס רְכִיבִים | 20.03.62 |
| computer performance evaluation | הַעֲרָכַת בִּצּוּעֵי מַחְשֵׁב | 20.05.15 |
| computer software configuration item | פְּרִיט תְּצוּרַת תָּכְנָה | 20.03.11 |
| computer-aided software engineering | הַנְדָּסַת תָּכְנָה בְּעֶזְרַת מַחְשֵׁב | 20.04.01 |
| concept phase | שְׁלַב הַהַמְשָׁגָה | 20.01.23 |
| conceptual system design | תֶּכֶן מַעֲרָכוֹת מֻשָּׁגִי | 20.03.04 |
| configuration control | בַּקָּרַת תְּצוּרָה | 20.08.08 |
| configuration control board | וַעֲדָה לְבַקָּרַת תְּצוּרָה | 20.08.09 |

| English Term | מונח עברי | מספר מונח |
|---|---|---|
| configuration diagram | תַּרְשִׁים תְּצוּרָה | 20.03.34 |
| configuration identification | זִהוּי תְּצוּרָה | 20.03.35 |
| configuration item | פְּרִיט תְּצוּרָה | 20.03.10 |
| configuration management | נִהוּל תְּצוּרָה | 20.03.36 |
| configuration status accounting | דִּווּחַ מִצַּב הַתְּצוּרָה | 20.03.39 |
| corrective maintenance | תַּחְזוּקָה מְתַקֶּנֶת | 20.06.04 |
| critical design review | תַּסְקִיר תְּכָן-קָרִיטִי | 20.03.54 |
| critical item | פְּרִיט קָרִיטִי | 20.03.13 |
| critical piece first | חֵלֶק קָרִיטִי תְּחִילָה | 20.03.15 |
| critical software | תָּכְנָה קָרִיטִית | 20.01.60 |
| cutover | הֶחְלֵף | 20.04.04 |
| data abstraction | הַפְשָׁטַת נְתוּנִים | 20.03.14 |
| data analysis | נִתּוּחַ נְתוּנִים | 20.02.18 |
| data inventory | מְצַאי נְתוּנִים | 20.01.26 |
| data-structure diagram | תַּרְשִׁים מִבְנֵה נְתוּנִים | 20.02.23 |
| data-structure-centered diagram | תַּרְשִׁים מְמֻרְכָּז מִבְנֵה-נְתוּנִים | 20.02.22 |
| decision table | טַבְלַת הַחְלָטָה | 20.07.18 |
| deliver (v) | מָסַר | 20.01.68 |
| deliverable | נִמְסָר | 20.01.69 |
| demodularization | אִחוּד מוֹדוּלִים | 20.03.26 |
| descendant | צֶאֱצָא | 20.01.67 |
| design document | מִסְמַךְ תְּכָן | 20.07.04 |
| design inspection | בְּחִינַת תְּכָן | 20.05.19 |
| design phase | שְׁלַב הַתְּכָן | 20.01.24 |
| design review | תַּסְקִיר תְּכָן | 20.03.51 |
| design unit | יְחִידַת תְּכָן | 20.03.09 |
| desk checking | בְּדּוּק יְדָנִי | 20.05.46 |
| detailed design | תְּכָן מְפוֹרָט | 20.01.27 |
| developmental baseline | קַו-בָּסִיס לְפִתּוּחַ | 20.07.10 |
| developmental configuration | תְּצוּרַת פִּתּוּחַ | 20.04.06 |
| developmental testing | בְּדִיקַת פִּיתּוּחַ | 20.05.28 |
| deviation | סְטִיָּה | 20.01.40 |
| diagnostic manual | מַדְרִיךְ אִבְחוּן | 20.07.26 |
| downsizing | שִׁדּוּד מַטָּה | 20.06.16 |
| downward compatible | תּוֹאֵם מַטָּה | 20.03.49 |
| downward compression | דְּחִיסָה מַטָּה | 20.03.27 |
| dynamic analysis | נִתּוּחַ דִּינָמִי | 20.05.22 |
| efficiency | נְצִילוּת | 20.01.41 |
| egoless programming | תִּכְנוּת לֹא-אָנוֹכִי | 20.04.07 |
| engineering change | שִׁינּוּי הַנְדָּסִי | 20.08.12 |
| engineering change proposal | הַצָּעַת שִׁינּוּי הַנְדָּסִי | 20.08.13 |
| entity-relationship diagram, E-R diagram | תַּרְשִׁים יֵשׁוּיוֹת-יַחֲסוֹת | 20.02.24 |
| entity-relationship model | מוֹדֵל יֵשֵׁיוֹת-יַחֲסוֹת | 20.02.27 |
| environmental requirement | דְּרִישָׁה סְבִיבָתִית | 20.02.07 |
| estimator | אַמְדָן | 20.02.29 |

| English Term | מונח עברי | מספר מונח |
|---|---|---|
| evaluation report | דּוּחַ הַעֲרָכָה | 20.07.19 |
| execution efficiency | נְצִילוּת בְּצוּעַ | 20.01.42 |
| extendability | כֹּשֶׁר הִתְרַחֲבוּת | 20.01.43 |
| factoring | פֵּרוּק לְגוֹרְמִים | 20.03.33 |
| fault threshold | סַף תַּקָּלוֹת | 20.05.63 |
| fault trace | עִקְבַת תַּקָּלוֹת | 20.05.62 |
| fault-rate threshold | סַף קֶצֶב תַּקָּלוֹת | 20.05.64 |
| feasibility study | חֵקֶר הִתָּכְנוּת | 20.02.02 |
| field-level sensitivity | רְגִישׁוּת בְּרָמַת הַשָּׂדֶה | 20.03.61 |
| form, fit and function | תְּצוּרָה כְּלָלִית | 20.03.37 |
| formal qualification review | תַּסְקִיר כְּשִׁירוּת פּוֹרְמָלִי | 20.05.13 |
| formal specification | מִפְרָט פּוֹרְמָלִי | 20.01.07 |
| formal testing | בְּדִיקָה פּוֹרְמָלִית | 20.05.29 |
| functional analysis | נִתּוּחַ תִּפְקּוּדִי | 20.02.16 |
| functional baseline | קַו-בָּסִיס תָּפְקּוּדִי | 20.07.11 |
| functional configuration audit | מִבְדַּק תְּצוּרָה תִּפְקּוּדִי | 20.05.20 |
| functional decomposition | פֵּרוּק תִּפְקּוּדִי | 20.03.31 |
| functional design | תָּכָן תִּפְקּוּדִי | 20.03.05 |
| functional requirement | דְּרִישָׁה תִּפְקּוּדִית | 20.02.08 |
| functional specification | מִפְרַט תָּפְקּוּדִי | 20.07.07 |
| functional testing | בְּדִיקָה תִּפְקּוּדִית | 20.05.27 |
| glass box | קֻפְסָה שְׁקוּפָה | 20.03.41 |
| hardware configuration item | פְּרִיט תְּצוּרַת חֲוּמְרָה | 20.03.12 |
| hierarchical decomposition | פֵּרוּק מִדְרָגִי | 20.03.32 |
| homogeneous redundance | יַתִּירוּת אֲחִידָה | 20.06.11 |
| implementation of a system | מִמּוּשׁ מַעֲרָכָת | 20.01.35 |
| implementation phase | שְׁלַב הַמִּמּוּשׁ | 20.01.34 |
| implementation requirement | דְּרִישַׁת מִמּוּשׁ | 20.02.09 |
| incremental development | פִּתּוּחַ תּוֹסַפְתִּי | 20.03.16 |
| informal testing | בְּדִיקָה לא-פּוֹרְמָלִית | 20.05.30 |
| information analysis | נִתּוּחַ מֵידָע | 20.02.17 |
| input-process-output | קֶלֶט-עִבּוּד-פֶּלֶט | 20.03.19 |
| installation and checkout phase | שְׁלַב (הַ)הַתְקָנָה וּבְדּוּק (הַ)שִׁחְרוּר | 20.01.33 |
| installation manual | מַדְרִיךְ הַתְקָנָה | 20.07.27 |
| instrumentation | מִכְשׁוּר | 20.06.13 |
| integrate | תִּכְלֵל | 20.03.59 |
| integrated infrastructure | תַּשְׁתִּית מְתֻכְלֶלֶת | 20.03.60 |
| integration test | בְּדִיקַת תִּכְלוּל | 20.05.31 |
| interface control | בַּקָּרַת מִנְשָׁק | 20.03.38 |
| interface requirement | דְּרִישַׁת מִנְשָׁק | 20.02.10 |
| interface testing | בְּדִיקַת מִנְשָׁק | 20.05.32 |
| lateral compression | דְּחִיסָה צִדִּית | 20.03.28 |
| lifecycle phase | שְׁלַב בְּמַחֲזוֹר חַיִּים | 20.01.20 |
| loopback testing | בְּדִיקַת לוּלָאָה | 20.05.34 |
| maintainability | תַּחְזוּקָתִיּוּת | 20.01.44 |

| English Term | מונח עברי | מספר מונח |
|---|---|---|
| maintenance | תַּחְזוּקָה | 20.06.01 |
| maintenance manual | מַדְרִיךְ תַּחְזוּקָה | 20.07.28 |
| menu by-pass | מַעֲקַף תַּפְרִיט | 20.03.48 |
| menu-driven | מֻנְחֵה תַּפְרִיטִים | 20.03.47 |
| metric | מֶטְרִיקָה | 20.05.06 |
| modular decomposition | פֵּרוּק לְמוֹדוּלִים | 20.03.30 |
| mutation testing | בְּדִיקַת מוּטַצְיָה | 20.05.35 |
| network chart | תַּרְשִׁים רֶשֶׁת | 20.08.06 |
| network planning | תִּכְנוּן רֶשֶׁת | 20.08.07 |
| object-oriented design | תֶּכֶן כְּוּוּן-עֲצָמִים | 20.01.65 |
| object-oriented design | תֶּכֶן כְּוּוּן עֲצָמִים | 20.03.07 |
| Open Information Model | מוֹדֵל מֵידַע פָּתוּחַ | 20.02.28 |
| operation and maintenance phase | שְׁלַב הַתִּפְעוּל וְהַתַּחְזוּקָה | 20.01.36 |
| operation testing | בְּדִיקַת תִּפְעוּל | 20.05.36 |
| operator manual | מַדְרִיךְ לַמַּפְעִיל | 20.07.30 |
| opportunity study | חֵקֶר הִזְדַּמְּנוּת | 20.02.01 |
| order clash | הִתְנַגְּשׁוּת סְדָרִים | 20.03.44 |
| packaging | אֲרִיזָה | 20.03.58 |
| parallel run | רִיצָה בְּמַקְבִּיל | 20.04.05 |
| parent | הוֹרֶה | 20.01.66 |
| path analysis | נִתּוּחַ נְתִיבִים | 20.02.19 |
| path condition | תְּנָאֵי נָתִיב | 20.02.20 |
| path expression | בִּטוּי נָתִיב | 20.02.21 |
| path testing | בְּדִיקַת נָתִיב | 20.05.37 |
| perfective maintenance | תַּחְזוּקָה מְשַׁכְּלֶלֶת | 20.06.05 |
| performance requirement | דְּרִישַׁת בִּצּוּעִים | 20.02.11 |
| performance testing | בְּדִיקַת בִּצּוּעִים | 20.05.38 |
| petri net | רֶשֶׁת פֶּטְרִי | 20.02.25 |
| physical configuration audit | מִבְדַּק תְּצוּרָה פִּיזִי | 20.05.21 |
| physical requirement | דְּרִישָׁה פִּיזִיקָלִית | 20.02.12 |
| pilot project | פְּרוֹיֶקְט חָלוּץ | 20.01.30 |
| pipe | צִנּוֹר | 20.03.56 |
| pipeline | צַנָּרֶת | 20.03.20 |
| pipelining | הַצְנָרָה | 20.03.57 |
| preliminary design | תֶּכֶן רִאשׁוֹנִי | 20.01.25 |
| preliminary design review | תַּסְקִיר תֶּכֶן-קְדָמִי | 20.03.52 |
| preventive maintenance | תַּחְזוּקָה מוֹנַעַת | 20.06.06 |
| problem definition | הַגְדָּרַת בְּעָיָה | 20.02.03 |
| product baseline | קַו-בָּסִיס לְמוּצָר | 20.07.12 |
| product specification | מִפְרָט מוּצָר | 20.01.08 |
| program maintenance manual | מַדְרִיךְ תַּחְזוּקַת תָּכְנִית | 20.07.25 |
| program mutation | מוּטַצְיָה שֶׁל תָּכְנִית | 20.05.60 |
| program network chart | תַּרְשִׁים-רֶשֶׁת שֶׁל תָּכְנִיּוֹת | 20.07.23 |
| program specification | מִפְרַט תָּכְנִית | 20.07.06 |
| programmer manual | מַדְרִיךְ לַתָּכְנִיתָן | 20.07.24 |

| English Term | מונח עברי | מספר מונח |
|---|---|---|
| project | פְּרוֹיֶקְט | 20.08.01 |
| project control | בַּקָּרַת פְּרוֹיֶקְט | 20.08.05 |
| project file | תִּיק פְּרוֹיֶקְט | 20.07.14 |
| project management | נִהוּל פְּרוֹיֶקְט | 20.08.02 |
| project plan | תָּכְנִית פְּרוֹיֶקְט | 20.07.15 |
| project planning | תִּכְנוּן פְּרוֹיֶקְט | 20.08.03 |
| project specification | מִפְרַט פְּרוֹיֶקְט | 20.08.04 |
| prototype | אַב-טִיפּוּס | 20.01.28 |
| prototyping | הַטְפָּסַת-אָב | 20.01.29 |
| qualification | כְּשִׁירוּת | 20.05.12 |
| qualification testing | בְּדִיקַת כְּשִׁירוּת | 20.05.39 |
| quality | אֵיכוּת | 20.05.01 |
| quality assurance | הַבְטָחַת אֵיכוּת | 20.05.02 |
| quality attribute | תְּכוּנַת אֵיכוּת | 20.05.04 |
| quality control | בַּקָּרַת אֵיכוּת | 20.05.03 |
| quality metric | מֶטְרִיקַת אֵיכוּת | 20.05.05 |
| rapid prototyping | הַטְפָּסַת-אָב מְהִירָה | 20.03.18 |
| rebase (v) | בִּסֵּס מֵחָדָשׁ | 20.03.64 |
| redundancy | יַתִּירוּת | 20.06.08 |
| reference book | סֵפֶר | 20.01.63 |
| regression testing | בְּדִיקַת תְּסוּגָה | 20.05.40 |
| requirement | דְּרִישָׁה | 20.01.05 |
| requirements analysis | נִתּוּחַ דְּרִישׁוֹת | 20.02.06 |
| requirements document | מִסְמַךְ דְּרִישׁוֹת | 20.07.13 |
| requirements phase | שְׁלַב הַדְּרִישׁוֹת | 20.01.22 |
| requirements review | תַּסְקִיר דְּרִישׁוֹת | 20.02.13 |
| resizing | שִׁדּוּד מֵחָדָשׁ | 20.06.17 |
| retirement phase | שְׁלַב הַזְּנִיחָה | 20.01.39 |
| reusability | רַב-שִׁמּוּשִׁיּוּת | 20.01.45 |
| reusable | רַב-שִׁמּוּשִׁי | 20.01.46 |
| reuse | שִׁמּוּשׁ חוֹזֵר | 20.01.64 |
| reversible execution | בִּצּוּעַ חוֹזֵר | 20.05.61 |
| right-sizing | שִׁדּוּד הוֹלֵם | 20.06.18 |
| robustness | חֹסֶן | 20.01.47 |
| self-descriptive | מְתֹאָר מֵאֵלָיו | 20.01.48 |
| self-documented | מְתֹעָד מֵאֵלָיו | 20.01.49 |
| serviceability | שֵׁרוּתִיּוּת | 20.01.50 |
| simulated configuration | תְּצוּרָה מֻדְמֵית | 20.03.63 |
| simulation | הַדְמָיָה | 20.01.32 |
| sizing | שִׁדּוּד | 20.01.51 |
| software design description | תֵּאוּר תְּכָן תָּכְנָה | 20.03.08 |
| software development cycle | מַחֲזוֹר פִּתּוּחַ תָּכְנָה | 20.01.04 |
| software development file, SDF [abb] | תִּיק פִּתּוּחַ תָּכְנָה | 20.07.16 |
| software development library | סִפְרִיַּת פִּתּוּחַ תָּכְנָה | 20.04.10 |
| software development plan | תוכנית-פרויקט לפרויקט פיתוח תוכנה. | 20.01.03 |

| English Term | מונח עברי | מספר מונח |
|---|---|---|
| software development process | תַּהֲלִיךְ פִּתּוּחַ תָּכְנָה | 20.01.02 |
| software diversity | קִדּוּד מִתְחָרֶה | 20.04.08 |
| software lifecycle | מַחֲזוֹר חַיֵּי תָּכְנָה | 20.01.62 |
| software package | חֲבִילַת תָּכְנָה | 20.01.61 |
| software product | מוּצַר תָּכְנָה | 20.01.14 |
| software quality metric | מֶטְרִיקַת אֵיכוּת | 20.05.65 |
| software requirements review | תַּסְקִיר דְּרִישׁוֹת תָּכְנָה | 20.02.14 |
| software requirements specification | מִפְרַט דְּרִישׁוֹת תָּכְנָה | 20.07.05 |
| Software test incident | אֵרוּעַ בדיקת תָּכְנָה | 20.05.53 |
| specification | מִפְרָט | 20.01.06 |
| specification change notice | הוֹדָעַת שִׁנּוּי מִפְרָט | 20.07.20 |
| spiral model | מוֹדֵל סְפִּירָלִי | 20.03.21 |
| standby redundancy | יַתִּירוּת בַּעֲתוּדָה | 20.06.10 |
| statement testing | בְּדִיקַת מִשְׁפָּטִים | 20.05.41 |
| static analysis | נִתּוּח סְטָטִי | 20.05.23 |
| stepwise refinement | עִדּוּן צַעַד-צַעַד | 20.03.17 |
| stress testing | בְּדִיקַת מַאֲמָץ | 20.05.42 |
| structural testing | בְּדִיקַת מִבְנֶה | 20.05.43 |
| structure clash | הִתְנַגְּשׁוּת מִבְנִים | 20.03.45 |
| structured design | תְּכֶן מֻבְנֶה | 20.03.06 |
| support manual | מַדְרִיךְ תְּמִיכָה | 20.07.29 |
| support software | תָּכְנַת תְּמִיכָה | 20.01.15 |
| system analysis | נִתּוּחַ מַעֲרָכוֹת | 20.02.15 |
| system data | נְתוּנֵי מַעֲרֶכֶת | 20.04.11 |
| system description | תֵּאוּר מַעֲרֶכֶת | 20.07.02 |
| system design | תְּכֶן מַעֲרָכוֹת | 20.03.01 |
| system design review | תַּסְקִיר תְּכֶן-מַעֲרֶכֶת | 20.03.53 |
| system development | פִּתּוּחַ מַעֲרָכוֹת | 20.01.01 |
| system development phase | שְׁלַב פִּתּוּחַ מַעֲרֶכֶת | 20.01.19 |
| system documentation | תִּעוּד מַעֲרֶכֶת | 20.07.01 |
| system follow-up | מַעֲקַב מַעֲרֶכֶת | 20.02.26 |
| system integration | תִּכְלוּל מַעֲרָכוֹת | 20.04.02 |
| system integrity | כְּלִילוּת מַעֲרֶכֶת | 20.05.10 |
| system lifecycle | מַחֲזוֹר חַיֵּי מַעֲרֶכֶת | 20.01.21 |
| system maintenance | תַּחְזוּקַת מַעֲרֶכֶת | 20.06.02 |
| system profile | פְּרוֹפִיל מַעֲרֶכֶת | 20.05.09 |
| system sizing | שִׁדּוּד מַעְרָכוֹת | 20.06.14 |
| system software | תָּכְנַת מַעֲרֶכֶת | 20.01.59 |
| system support | תְּמִיכָה בַּמַּעֲרֶכֶת | 20.01.38 |
| system testing | בְּדִיקַת מַעֲרֶכֶת | 20.05.44 |
| test case specification | מִפְרַט מִקְרֵה בְּדִיקָה | 20.07.08 |
| test driver | הֶנַּעַ בְּדִיקָה | 20.05.56 |
| test generator | מְחוֹלֵל בְּדִיקוֹת | 20.05.55 |
| test incident report | דּוּחַ אֵרוּעַ בְּדִיקָה | 20.07.21 |
| test phase | שְׁלַב הַבְּדִיקָה | 20.01.31 |

| English Term | מונח עברי | מספר מונח |
|---|---|---|
| test plan | תָּכְנִית בְּדִיקָה | 20.07.17 |
| test readiness review | תַּסְקִיר מוּכָנוּת לִבְדִיקָה | 20.05.14 |
| test set architecture | אַרְכִיטֶקְטוּרַת עֶרְכַּת בְּדִיקָה | 20.05.54 |
| testability | בֶּדִיקָתִיּוּת | 20.01.52 |
| testing | בְּדִיקָה | 20.05.24 |
| thrashing | דִּשְׁדּוּשׁ | 20.06.12 |
| timing | תִּזְמוּן | 20.01.53 |
| timing analyzer | נַתַּח תִּזְמוּן | 20.01.54 |
| top-down | מַעְלָה-מַטָּה | 20.01.16 |
| traceability | עֲקִיבוּת | 20.05.07 |
| traceability matrix | מַטְרִיצַת עֲקִיבוּת | 20.03.42 |
| transaction | תְּנוּעָה | 20.03.23 |
| transaction analysis | נִתּוּחַ תְּנוּעוֹת | 20.03.24 |
| transaction matrix | מַטְרִיצַת תְּנוּעוֹת | 20.03.43 |
| transform analysis | נִתּוּחַ הַתְּמָרָה | 20.03.25 |
| unit test | בְּדִיקַת יחידה | 20.05.51 |
| unit testing | בְּדִיקַת יְחִידָה | 20.05.45 |
| upsizing | שִׁדּוּד מַעְלָה | 20.06.15 |
| upward compatible | תּוֹאֵם מַעְלָה | 20.03.50 |
| upward compression | דְּחִיסָה מַעְלָה | 20.03.29 |
| usability | שִׁמּוּשִׁיּוּת | 20.01.55 |
| usability test | בְּדִיקַת שִׁמּוּשִׁיּוּת | 20.05.52 |
| user documentation | תִּעוּד מַשְׁתַּמֵּשׁ | 20.07.03 |
| user interface | מִנְשָׁק מִשְׁתַּמֵּשׁ | 20.01.56 |
| user manual | מַדְרִיךְ לַמִּשְׁתַּמֵּשׁ | 20.07.31 |
| user portability | יְבִילוּת מִשְׁתַּמֵּשׁ | 20.01.57 |
| utilization | נִיצּוּל | 20.01.58 |
| validation | תִּקּוּף | 20.05.50 |
| variant | נֹסַח אַחֵר | 20.05.59 |
| verification | אִמּוּת | 20.05.49 |
| version | גִּרְסָה | 20.01.37 |
| version description document | מִסְמַךְ תֵּאוּר גִּרְסָה | 20.07.22 |
| waiver | כְּתַב וִתּוּר | 20.08.14 |
| walk-through | בִּיקֹּרֶת נוֹקֶבֶת | 20.03.55 |
| waterfall model | מוֹדֵל מַפַּל-מַיִם | 20.03.22 |
| workload | עֹמֶס עֲבוֹדָה | 20.05.16 |
| workload model | מוֹדֵל עֹמֶס עֲבוֹדָה | 20.05.17 |