Functional testing ensures that functions and features of the application work properly.

"מה" מערכת אמורה לעשות

Nonfunctional testing examines other aspects of how well the application works

"איך" מערכת אמורה לעשות

Positive testing determines that your application works as expected

Negative testing **ensures that your application can gracefully handle invalid input or unexpected user behavior**

**מסמך אפיון**

**תכנון ועיצוב המערכת לפי דרישת הלקוח**

מסמך דרישות - "מה לקוח מצפה לקבל

מסמך אפיון - "איך נממש את מה שלקוח מצפה לקבל

דרישות + אפיון = מה המערכת אמורה לעשות

בדיקת שפיות

מדובר על בדיקות מאוד בסיסיות, שיכולות להעניק את המידע האם ניתן להיכנס לבדיקות מעמיקות יותר או לא

האם מערכת מסוגלת לבצע תפקוד בסיסי

בדיקות עשן

למצוא בעיות קריטיות בהקדם האפשרי

רגרסיה

אם תיקון התקלה מסויימת יצר תקלות חדשות במקומות שקודם תפקדו כראוי?

ממשק  משתמש : האם רכיב מסוים עובד כפי שצריך לעבודGUI

בדיקת אלמנטים במסך

הי בדיקה   שמבהירה לנו  האם מה שהוגדר באפיון  אכן מתממש במציאות בכל הנוגע  להתנהגות  שדות (של  הזנת מידע) ותפקודם.  למעשה  כאן אנו בודקים את החוקיות

של השדות  כדי שמשתמש הקצה יוכל  לתפעל את המערכת בקלות ומאידך  גם אם משתמש הקצה יעשה טעויות במילוי השדות  שהמערכת לא  תאפשר לו.

וכמובן בודקים האם כאשר מלאתי שדה אחד כהלכה האם  האם השדה/המסך הבא יפתח לי

* האם  אני יכול  להשים אותיות במקום של מספרים  בשדה שיועד למספרים
* האם  הקשה    לא   נכונה של  פרטים  בשדה אחד  תאפשר לי  להיכנס לשדה  הבא  הנדרש-שלילי

בדיקת תקינות פונקציונאלית

תהליך לא מלא

רק שמאזינים כל הנתונים דרושים ,רק אז עובר למסך הבא

בדיקת תהליכים : תהליך עסקי מלא

 זה  שאני  נכנס למערכת  של דואר אלקטרוני  -   כניסה   לחשבון מייל(תהליך  אחד)  ושליחה ממנו(תהליך שני)  זה  בדיקת  תהליכים.

בדיקת קצה לקצה -(תקינות פונקציונלית) – רק שממלא כל הנתונים דרושים ,רק אז עובר למסך סופי- רצף תקין

**. Integration – בדיקת אינטגרציה**

**לבדוק את ההשפעה שיש לחלק מוגדר במערכת עם חלקים אחרים(שקע חשמלי,.**

**ממשקים -Interface**

**בדיקת הממשקים הפנימיים לבין הממשקים חיצוניים**

**(תשלום לסופר דרך חברת אשראי)**

האם המערכת יכולה להתמודד עם  ממשקים  שנכנסים   ולהשתמש בנתונים בקלות וגם אם המערכת יכולה  להוציא נתונים  והם יתקבלו  במערכת  אחרות  בארגון  כמו  בדוגמא של הנהלת החשבונות

בדיקתהרשאות **:**

אם אדם עם ההרשאה המתאימה קרי עם שם וסיסמא תקפים יכול להיכנס למערכת והאם להבדיל, משתמש עם סיסמא או שם משתמש לא תקין יוכל אף הוא להיכנס למערכת( בנקאי,לקוח,מנהל)

:SECURITYבבדיקות אבטחה\

מול נסיונות חדירה אליה

USER STORY:

מקרה\סיפור שימוש, פעולה עסקית שלמה מתחיתה עד הסוף

בדיקות שרידות והתאוששות

האם מערכת תשחזר מידע בהתאם לנקודת השחזור אחרון

בדיקות שימושיות

* "בודקים מקצועניים" – אלפא ,בטאUsability

UI user interface

וויזואליות / נראות המערכת ,

כל הקשור בממשק המשתמש כגון: גודל וצבע גופן, צבעי רקע, שדות

מסגרות תוכן, תמונות, כפתורים וכדומה.

UX user expireance

כמה זמן לוקח למשתמש חדש ללמוד את שימוש במערכת

ניווט במערכת – כמה קל לנווט בתוך המערכת

בדיקות אלפא/בתא

בדיקות שמבוצעות על ידי קבוצת משתמשים מוגדרת סגורה(אלפא)

או פתוחה (בתא) שמדווחת על תקלות הצפות תוך כדי שימוש שוטף.

בדיקת הבטחה- **Security**

אם פרוטוקול מאובטח

https

האם תעודות של אתר בתוקף

**סודיות, שלמות המידע, אימות זהות הרשאות. דגש על אבטחת מידע ובדיקת העמידות של המערכת בפני התקפות זדוניות**

בדיקות ביצוע ועומסים

**Performance – בדיקת ביצועים**

**בדיקת תפקוד רכיבי המערכת במצב של פעילות מלאה ולאורך זמן (נקודת שבירה).**

בסוג בדיקות זה נבדקת יכולת התגובה של צד השרת במערכות שרת/לקוח בהן צפויים משתמשים רבים בו זמנית

* נפח – פעולה מערכת כאשר Volume

מלא יותר מ- % 90 - DB

היא בדיקה שבודקת את התנהגות וגבולות המערכת תחת הצפת המערכת בנתונים וכמו בבדיקה הקודמת – בודקת את קצוות המערכת בשימוש בכמויות אדירות של מידע

.LOAD

עומס – בודקים אם המערכת עובדת עם מספר משתמשים רגיל ותחת זרימת נתונים נורמטיבית

* - לחץSTRESS
* בודקים אם המערכת עובדת עם מספר משתמשים הגבוה בהרבה מהרגיל וכמו כן שימוש במידע רב, תוך גרימת זעזועים (לדוגמה: ניתוק וחיבור משתמשים, שימוש וביטול פונקציונליות במערכת)

הסבת נתונים

כאשר מחליפים מערכת או משדרגים למוצר אחר, נדרשות ביצוע בדיקות הסבה מקיפות כדי לא לייצר נזק לתהליכים העסקיים באירגון

.

**בדיקת תחזוקתיות**

בדיקה שבודקת שאפשר לעדכן או לתקן את התוכנה אחרי הוצאתה לאור, בדיקה זו בודקת גם אם הקוד כתוב בצורה פשוטה.

בדיקות גיבוי  ושחזור

Backup and recovery tests

בדיקות התקנה/הסרה

תקינות ערכת ההתקנה

## **Compatibility** / בדיקות תאימות – בדיקת תאימות האתר לסביבות שונות.

WEB כמעט וכל מוצר שמפותח היום הוא

ולכן, חובה שתהיה לו תאימות להמון דפדפנים שונים, הן על הדסקטופ והן על המובייל. בעזרת בדיקות אלו, מוודאים שהליך זה באמת מתחרש. כך שלא יקרה תסריט בו לא תהיה תאימות לדפדפן, מערכת הפעלה או תוכנות אחרות שאמורות להתממשק איתן

**אתר בדפדפנים שונים כמו:**

**Explorer , Edge , Firefox , Google Chrome**

**פלטפורמות שונות לדוגמא:**

**Windows, Linux וכו'.**

**Accessibility / בדיקות נגישות –**

כמה המוצר נגיש עבור בעלי מוגבלויות, כשכידוע, יצא חוק לא מזמן שמחייב כל בעל עסק או ארגון שהמוצרים שלו יהיו ניגשים עבור עיוורים, כבדי שמיעה ועד שאר מוגבלויות אחרות

**בדיקות בינלאומיות**  - בדיקות המתמקדות בשימוש בתוכנה בממשקים בשפות שונות. למשל – אתרי אינטרנט רבים (כמו [ויקיפדיה](https://he.wikipedia.org/wiki/%D7%95%D7%99%D7%A7%D7%99%D7%A4%D7%93%D7%99%D7%94" \o "ויקיפדיה)) בהם יש דפים בשפות שונות, מבצעים בדיקות מהסוג הזה

בדיקות  התאוששות   מנפילות

**בדיקה לאחר שהמערכת נפלה בכדי לראות שכל הנתונים שרדו לאחר נפילת המערכת.**

בדיקות התאוששות מנפילות מראות לנו מה קורה כאשר כאשר יש תקלה בלתי צפויה במחשב או בשרת  שעליו נמצאת התוכנה האם המידע  אובד לנו  ,האם חל בו שינוי כלשהו

אבטחת מערכות מידע :

[**זמינות**](https://he.wikipedia.org/w/index.php?title=%D7%96%D7%9E%D7%99%D7%A0%D7%95%D7%AA&action=edit&redlink=1)**(Availability)**

המידע יהיה זמין ומוכן לשימוש. שרת שנפל הוא דוגמה לפגיעה בזמינות המידע.

[**סודיות**](https://he.wikipedia.org/wiki/%D7%A1%D7%95%D7%93%D7%99%D7%95%D7%AA) **-Confidentiality**

המידע יהיה נגיש רק למי שמורשה לצפות בו או לעשות בו שינויים

[**שלמות מידע**](https://he.wikipedia.org/w/index.php?title=%D7%A9%D7%9C%D7%9E%D7%95%D7%AA_%D7%9E%D7%99%D7%93%D7%A2&action=edit&redlink=1)**Integrity**

מידע המתקבל שלם ולא עבר שינויים במהלך שליחתו

[אימות זהות](https://he.wikipedia.org/wiki/%D7%90%D7%99%D7%9E%D7%95%D7%AA_%D7%96%D7%94%D7%95%D7%AA) (Authentication

היה ניתן לאמת את זהות השולח בצורה מפורשת וניתן יהיה לאתר את מקור המידע

* **Software Test Plan –**[**STP**](https://www.webforalls.co.il/%d7%9e%d7%a1%d7%9e%d7%9a-stp/)
* **מסמך שנכתב כבר בשלב מסמך הדרישות או האפיון, תכנית על של הבדיקות מה ייבדק לפי תעדוף, כולל ניהול סיכונים ומשאבים, לא כולל תסריטים**

**Software Test Description –**[**STD**](https://www.webforalls.co.il/%d7%9e%d7%a1%d7%9e%d7%9a-std/)

* תכנון הבדיקה. תכנון זה יכלול את תהליך הבדיקה והתוצאות המצופות. STD
* **תכנית בדיקות מפורטת, כוללת את כל התסריטים בכל רמה ובכל סוג, לרבות צעדים, תנאים מקדימים להרצה ותוצאות רצויות**
* **.**
* **Software Testing Results-**[**STR**](https://www.webforalls.co.il/%d7%9e%d7%a1%d7%9e%d7%9a-str/)**– מסמך סיכום בדיקות**
* **מה הבדיקות שבוצעות והסטטוס שלהן. סך התקלות לפי רמות חומרה. פירוט התקלות הקריטיות שיכולות לסכן את המערכת, קביעה אם המערכת עמדה ביעדיה, ערכת הסיכונים, על פי מסמך זה הלקוח מחליט אם לצאת לייצור או לא.**

Test case –

מקרה בדיקה זה תהליך אשר מורכב משלבים שיש לעבור לצורך ביצוע משימה, כאשר לכל שלב יש תיאור מה צריך לעשות .לאחר ביצוע כל שלב יש לבדוק האם יש התאמה לתוצאה צפויה ,כאשר יש התאמה – עבר,אם לא – נכשל

בדיקת תרחיש  - - Scenario testing

 נעשה שימוש בסיפורים היפותטיים על-מנת לסייע לבודק לחשוב על בעיה מורכבת או על מערכת מורכבת. תרחישים אלו לרוב אינם נכתבים בפירוט. התרחישים יכולים להיות פשוטים כמו דיאגרמה של סביבת בדיקה, או מורכבים כמו פרוזה. בדיקת התרחיש האידיאלית הוא סיפור מניע, אמין, מורכב, וקל להערכה.

בדיקות התרחיש בדרך-כלל נבדלות ממקרי הבדיקה בכך שמקרי הבדיקה הם צעדים בודדים בעוד תרחישים מכסים מספר צעדים.

**תסריט בדיקה**

 הוא רשימת צעדים קצרה או ארוכה, מאורגנת לפי שלבים עוקבים

[**בדיקות יחידה**](https://he.wikipedia.org/wiki/%D7%91%D7%93%D7%99%D7%A7%D7%95%D7%AA_%D7%99%D7%97%D7%99%D7%93%D7%94) (Unit)

בדיקות ברמת יחידת תוכנה (מודול). לרוב מבוצעות על ידי מפתח התוכנה

[**בדיקות אינטגרציה**](https://he.wikipedia.org/wiki/%D7%91%D7%93%D7%99%D7%A7%D7%95%D7%AA_%D7%90%D7%99%D7%A0%D7%98%D7%92%D7%A8%D7%A6%D7%99%D7%94) (Integration)

בדיקת שילוב יחידות תוכנה בהיקפים שונים, החל משתי יחידות ועד לכלל היחידות במערכת.

* System
* בדיקות מערכת

בדיקות המערכת – בודקות את התנהגות המערכת כמכלול. בדיקות בהיקף גדול, פונקציונאליות ולא פונקציונאליות בעיקר, שמטרתן לוודא כי המערכת עושה את מה שהיא נדרשת לעשות, ולא עושה מה שאינה נדרשת. הבדיקות תכלולנה בין היתר: בדיקות מסדי נתונים, קשרי גומלין בין תהליכים, קשרים עם מערכות חיצוניות, בדיקות ביצועים, בדיקות עומסים, שרידות, שימושיות, בדיקות אבטחת מידע, ובדיקות התאוששות ממצבי כשל

Acceptance test

בדיקות קבלה - – שלב הבדיקות הסופי – לרוב מבוצע ע"י הלקוח\ המשתמש -/– בדיק ות מוכנות המערכת

. בדיקות מוכנות המערכת

מטרתן לוודא כי המערכת פועלת בהתאם לדרישות, לבסס בטחון במערכת, ולוודא כי היא מוכנה לעלייה לאוויר

בדיקות נסיגה

לאִימות פעילות המערכת לאחר שבוצעו בה שינויים. לוודא שמה שעבד לא התקלקל בעקבות העברת גרסה

 (White Box)

בדיקות אלו מתבססות על הכרת [קוד המקור](https://he.wikipedia.org/wiki/%D7%A7%D7%95%D7%93_%D7%9E%D7%A7%D7%95%D7%A8) של התוכנה ובניית תוכניות בדיקה המותאמות לנתיבי הזרימה האפשריים של הקוד. בדיקות יחידה עשויות להיות בדיקות מסוג קופסה לבנה. בסוג בדיקות זה, על הבודק להכיר את הלוגיקה של הקוד, וכן, עליו להיות בעל ידע בשפת התכנות בה כתובה התוכנה.

Black Box

דיקות אלו אינן מכירות את המבנה הפנימי של המערכת ומתבססות על בדיקת הפלט הצפוי לקלט מסוים בהתאם לתכנון מוקדם כלשהו. בדיקות קבלה מתבצעות בשיטה זו בדרך כלל. בסוג בדיקות זה, הבודק חייב לדעת את פירוט דרישות המערכת, וכן, עליו לדעת לאיזה פלט מהתוכנה עליו לצפות עבור קלט מסוים. עם זאת, הבודק **אינו** חייב להכיר את הלוגיקה של הבעיה או אפילו לדעת את שפת התכנות בה היא כתובה

Gray Box

בדיקות אלו מכירות במבנה הפנימי של המערכת אך משתמשות בידע זה על מנת לבצע בדיקות בסגנון קופסה שחורה. כך לדוגמה שינוי של מאגרי הנתונים לבדיקת פלט מסוים או שימוש ב[הנדסה הפוכה](https://he.wikipedia.org/wiki/%D7%94%D7%A0%D7%93%D7%A1%D7%94_%D7%94%D7%A4%D7%95%D7%9B%D7%94) על מנת לאתר את גבולות הפעולה של המערכת, אלו הן דוגמאות נפוצות לבדיקות בשיטה זו.

EPIC

נושא רחב (דרישה גדולה ביותר) – למשל

או 2) כספומטBooks1)

FETCHER

יכולת (דרישה בגודל בינוני) – למשל :

משיכת כסף( 2 Add new book (1

USER STORY

סיפור משתשמש (דרישה הקטנה ביותר) –איך משתמשים יכולים להשתמש ביכולת

למשל

Liberian can add new book using specific screen( 1

משתמש עם יתרה מסוימת מנסה למשוך סכום עולה על היתרה(2

**חומרות באג**

* קריטית (**Critical**)–
* מערכת או אפלקציה קורסת ולא עובדת כלל , אין דרך לעקוף ,מוצר לא שווה כלום ללא תיקון

משמעותית (**Major**)

אפלקציה או רכיב משמעותי לא עובדים כמצופה ,יש דרך לעקוף את זה

מזערית (**Minor**)–

תפקוד משני לא עובד כמצופה

טריוויאלי/קוסמטי (**Trivial/Cosmetic**)

שגיאת כתיב ,חוסר עקביות במראה ,שיפורים

**עדיפות נפוצות של באגים:**

דחוף (**Urgent**) – תקן מיד

גבוה (**High**)– תקן בהקדם האפשרי

נורמלי (**Normal**)– תקן בשחרור הבא

* נמוך (**Low**)– תקן כשיהיה זמן פנוי.
* - Exploratory testing
* בדיקות חוקרות היא גישה לבדיקות תוכנה אשר באופן תמציתי מתוארת כלימוד המערכת ללא תיעוד, באופן סימולטני, עיצוב בדיקה וביצוע בדיקה
* STLC

The Software Testing Life Cycle (STLC) is a **sequence of specific actions performed during the testing process to ensure that the software quality objectives are met**

* Software Development Life Cycle SDLC
* Requirement->analysis->design->development->testing->deployment/maintenance

מודל סדרתיים שלב חדש מתחיל כששלב קודם מסתיים ,אין חפיפה

מתחיל רק אחרי שפיתוח מסתיים Waterfall

מודל סבבים – בכל סבב נמסרת תוכנה עובדת

מודל מצטבר- נעשה בשלבים

* -דגש רב על יכולת התגובה לשינוי, יעילות ואיכותAgile

סוגי מערכת:

מערכת מידע-שליפה ,הצגה ,עידכון של נתונים

זמן אמת – בגבלת זמן

מערכת שליטה ובקרה-

תוכנת מדף-

בתחזוקה –