

רמות בדיקה	סוגי בדיקה	טכניקות בדיקה
<b>UNIT</b> <p>נועדה לבדוק תקינות של חלק ספציפי בקוד (פונקציה) נעשית ע"י צוות פיתוח.</p>	<b>FUNCTIONAL</b> <p>נועדה לוודא את תקינות המערכת בהיבטים הבאים: התנאים שקיימים במערכת, הפונקציות החשובות, הייעוד העסקי, אימות הנתונים ב-DB והלוגיקה והקשרים בין מסך ולמסך במוצר.</p>	<b>NEGATIVE</b> <p>להזין קלט לא תקין/ לא צפוי.</p>
<b>INTEGRATION</b> <p>נועדה לבדוק תקינות של אינטגרציה - תקשורת של חלקים בקוד אחד עם השני. (פונקציות שמתקשרות או מגיבות אחת לשנייה) נעשית ע"י צוות פיתוח.</p>	<b>ERROR HANDLING</b> <p>שימוש בטכניקה שלילית על מנת ליצור שגיאות צפויות ולא צפויות לפי מסמך האפיון. לוודא איך המערכת מתנהגת בכל סימולציה וששום דבר לא קורס.</p>	<b>BOUNDARY</b> <p>מתמקדת בבדיקת הגבולות או הערכים הקיצוניים של שדה הקלט. המטרה של טכניקה זו היא לזהות בעיות שעלולות להתרחש כאשר הנתונים נמצאים בקצה הטווח המקובל שלהם. נועדה לחסוך זמן.</p>
<b>SYSTEM</b> <p>- <b>SMOKE</b> בדיקה כללית על הפונקציות והתהליכים קריטיים במערכת, לא מעמיקה.</p> <p>- <b>SANITY</b> בדיקה ספציפית יותר מעמיקה על פיצ'ר ספציפי או תיקון באג ספציפי.</p> <p>- <b>REGRESSION</b> בדיקה רחבה יותר ומעמיקה משפיות ועשן, נועדה לבדוק שינויים שנכנסו בקוד לא גורמים לתקלות במקום אחר בתוכנה והשינויים שהוכנסו עובדים כמצופה. <u><b>בודקים הכל גם את החלקים עם השינויים וגם בלי השינויים</b></u></p>	<b>GUI</b> <p><b>Graphical User Interface</b>  <u>אלמנטים חזותיים</u> - צבע גודל ומיקום תקינים של מסכים/פקדים/שדות.  <u>אלמנטים אינטראקטיביים</u> - וידוא תקינות של כל רכיבי הממשק שאמורים להגיב לאינטראקציה עם המשתמש. (הנפשות, לחצנים שמגיבים לעכבר מעליהם)  <u>שדות חובה תקינים</u>  <b>הגנה</b> - בודקת שאין שגיאות כתיב, טקסט שגוי וכדומה.</p>	<p>תחת SYSTEM ולפני SMOKE <u>לאחר וידוא שסביבת העבודה זהה ככל האפשר לסביבת המשתמש/לקוח</u></p> <p><b>Risk Based Testing</b>  כשאנחנו מתחילים לבדוק את המערכת שלנו אנחנו נרצה לעשות איזשהו תיעודף (במיוחד אם אנחנו קצרים בזמן). במקומות "רגישים" וקריטיים במוצר.</p> <p><u>למשל: לוודא מול מפתח שעשה UNIT TESTING</u>, לבחון ביתר תשומת לב פיצ'ר שפיתח מפתח עם פחות ניסיון או פיצ'ר שמורכב מהרבה פונקציות שונות.</p>
<b>ACCEPTANCE</b> <p>מתבצע ע"י משתמשי קצה או מנהל מוצר ולפעמים שוכרים בודקים חיצוניים לזה. (בעסקים - B2B - בדר"כ משתמשי קצה) מעריכה האם התוכנה עומדת בדרישות העסקיות ומוכנה להתקנה ב-"ייצור" Production.</p> <p>נועדה ללקוח (לא ליוצר) תחושת ביטחון שהמוצר שהוא עומד להתקין אכן עומד בהצלחה בדרישות שהוא דרש.</p>	<b>Compatibility</b> <p>בדיקת תאימות מתמקדת בלוודא שהתוכנה עובדת בהתאם לדרישות ומכשירים ופלטפורמות שונות. <u>לדוגמא: מערכות הפעלה שונות, מכשירים שונים, דפדפנים שונים.</u></p>	<p><b>בדיקות חקר</b> <b>Exploratory Testing</b>  בדיקות שלא כותבים להם תרחישי בדיקה מוכנים מראש. אלא הבדק יוצר אינטראקציה עם התוכנה תוך שהוא משתמש בידע, בניסיון ובאינטואיציה שלו/ה.</p> <p><u>הינן בדיקות משלימות ולא משתמשות כתחליף לבדיקות מתוכננות.</u></p> <p>דגשים חשובים הם:  - הצבת יעדים (באגים, זמני תגובה, שימושיות וכדומה)  - הקצבת זמנים</p>

<p><b>בדיקות קבוצות שקילות</b> <b>Equivalence Partitioning</b></p> <p>עוזרת לנו לצמצם את מספר מקרי הבדיקה. הנחת העבודה היא שאם ערך אחד מתנהג בצורה מסוימת אז גם ערכים אחרים מאותה קבוצה יתנהגו דומה.</p> <p>לדוגמה: בדיקת שדה טקסט בו מקבלים גיל מהיוזר וערכים ואליריים הם רק בין 20-65 יהיו לנו 3 קבוצות שקילות</p> <p>Invalid Partitioning - טווח גילאים 0-20 מקרי קצה שליליים מלמטה</p> <p>Valid Partitioning – טווח גילאים 21 – 65 מקרי קצה חיוביים</p> <p>Invalid Partitioning - טווח גילאים 66- ומעלה מקרי קצה מלמעלה</p>	<p><b>Usability</b></p> <p>מתמקדת בחווית המשתמש והיא מתבצעת לאור מסמכים והנחיות שכתבו אנשי ה-UX. המטרה העיקרית שלה היא להעריך עד כמה התוכנה קלה לשימוש ועד כמה היא ידידותית למשתמש.</p>	
<p><b>טכניקות בדיקות תהליכים עסקיים</b> <b>E2E Business process</b></p> <p>טכניקת E2E היא טכניקת בדיקה שמאמתת את ה"זרימה של התהליך העסקי" מתחילתו ועד סופו.</p> <p>לדוגמה: <u>אפליקציות של בנקים -</u> flow של בדיקת עו"ש, flow של ביצוע העברה, המרת מט"ח</p> <p><u>אפליקציות של אתרי קניות -</u> flow של קנייה של סט מוצרים מאוד ספציפי (למשל שיתוף פעולה בין אתר קניות לחברת מוצרי טיפוח beauty). Flow של תשלום באמצעות כרטיס אשראי מאסטרקארד (פיצ'ר חדש שנכנס שהמנהל מוצר רוצה שתבדקו).</p>	<p><b>Accessibility</b></p> <p>נגישות, לא אנחנו נבצע יש חברות מיוחדות המוסמכות לכך. מתמקדת בהערכה האם התוכנה נגישה לאנשים עם מוגבלויות</p>	
<p><b>טכניקת בדיקות סטטיות</b> <b>Static Testing</b></p> <p>בדיקה סטטית זה כל בדיקה של התוכנה ללא הפעלתה בפועל.</p> <p><b>בדיקה של בקרת איכות</b> <b>לדוגמה:</b> בקרה על מסמכים</p> <p>- סקירה של מסמך איפיון בשלב מוקדם וזיהוי שחסרה בו התייחסות מסוימות לדוג' התייחסות למבנה נתונים, אספקטים של אבטחת מידע או להתמשקות למערכות חיצוניות ועוד.</p> <p>- סקירה של מסמך איפיון וזיהוי שהדרישות בו לא ברורות מספיק – לדוג' למי יש הרשאות לבצע אילו פעולות.</p> <p>- סקירה של מסמך האפיון וזיהוי שקיימות דרישות סותרות שלא יכולות להתקיים ביחד.</p>	<p><b>Localization</b></p> <p>בדיקה שהמוצר מותאם למדינה בה הוא מותקן</p> <p><b>מימושים:</b> שפה, מטבעות, תאריכים בהתאם לאזור זמן, קידומות טלפון ומיקוד, ערים ואזורים מתאימים למדינה, רגולציות מקומיות ושיקולים תרבותיים כמו איסור מכירה של מוצר מסוים מגיל מסוים וכדומה.</p>	

	<p><b><i>Install/Un-Install</i></b></p> <p>בדיקת התקנה/הסרה היא חיונית כדי לספק למשתמשים חוויה חלקה וללא שגיאות. תהליך ההתקנה של תוכנה הוא בעצם המפגש הראשון של המשתמש עם המוצר. בעיות או סיבוכיות בתהליך ההתקנה עלול לגרום למשתמש תחושה ראשונית שלילית שיהיה קשה לשנותה לאחר מכן.</p>	
	<p><b><i>Interface (Integration)</i></b></p> <p>בדיקה של אינטגרציה (תקשורת ותגובה) בין שתי מערכות או יותר. <u>למשל</u>: התחברות עם חשבון גוגל לפייסבוק, פייפאל לשלם דרך אתרים. ווצאפ עם המצלמה, וויז, גלריה, אנשי קשר, יוטיוב, קבצים.</p>	
	<p><b><i>Recovery</i></b></p> <p>הקרסה מכוונת של התוכנה. לבדוק <u>האם</u> היא מתאוששת, <u>כמה מהר</u> מתאוששת ו<u>באיזה מצב</u> היא נמצאת לאחר ההתאוששות.</p>	
	<p><b><i>סוג בדיקה - בדיקת ביצועים</i></b></p> <p><b><i>Performance Testing</i></b></p> <p><u>Load Testing</u> - בדיקת עומסים, בדיקה כיצד מתנהגת התוכנה בתנאי עומס צפויים</p> <p><u>Stress Testing</u> - "מאמץ" נועדו לבדוק איך המערכת מתפקדת בתנאי עומס לא צפויים</p>	