

אימות והוכחת תקפות - V&V

המדריך

מסמך זה הוא המדריך הראשי והמלא של נושא אימות והוכחת תקפות V&V (Verification & Validation), או בשמו האחר "תורת הבדיקות". זהו אוסף פעולות, שיטות וכלים המקדמים את בדיקת המערכת, כולל שלב הבדיקות אך הרבה מעבר לו. הדגש כאן הוא על מכלול הבדיקות המתבצעות לאורך מחזור החיים של הפרויקט החל מאפיון המערכת, דרך שלב עיצוב ובנייה, התקנה והרצה, תפעול ותחזוקה, כולל שלב הבדיקות עצמו, כמובן. אימות והוכחת תקפות מקיפים את הפרויקט כולו ואינם מתמצים בשלב או תת שלב מסוים במחזור החיים, או ברכיב מסוים בעץ המערכת.

מדריך זה הוא "תורת" ו"מניפת" ומטרתו להציג את הנושא בכללותו ולשמש הפנייה, משולבת בדברי הסבר, לקיטים המעשיים העוסקים במקטעים השונים של בדיקת המערכת, כגון: אפיון (הכנה ראשונה של תכנית הבדיקה), עיצוב ובנייה (בדיקות יחידה), שלב הבדיקות (בדיקות מערכת ובדיקות קבלה), התקנה והרצה (בדיקות מוכנות והתקנה) ועוד.

תוכן העניינים

2	מבוא - הסבר כללי
4	סוגי בדיקות
8	מימוש סוגי הבדיקות
10	שלב הבדיקות
12	לאורך מחזור החיים
14	עץ המערכת
16	הרחבות
17	טכניקות וכלים

©

נוהל מפת"ח הוא מוצר המוגן בזכויות יוצרים
 הזכויות במגזר הממשלתי הן של משרד האוצר
 הזכויות מחוץ למגזר הממשלתי הן של מתודה מחשבים בע"מ
 זכויות השימוש של רוכשי הנוהל הן בהתאם לרישוי שברשותם.

מבוא - הסבר כללי

סוגים שונים של בדיקות מבוצעים לאורך מחזור חיי המערכת. שלב מיוחד במחזור חיים – שלב הבדיקות – מוקדש כולו לנושא ובו נבדקת המערכת "לעומק ולרוחב" על כל היבטיה במטרה לוודא מוכנותה להעברה לתפעול וייצור. אך אין זה השלב היחיד במחזור החיים בו יש התייחסות לנושא הבדיקות. בשלב האפיון, למשל, נבנית תכנית בדיקה ראשונה ומוערכים היקף וסוג הבדיקות שהמערכת תצטרך לעבור בהמשך הדרך. זו פעולת בדיקות מובהקת המתבצעת כחלק מהאפיון. בשלב עיצוב ובנייה מתוכננות ומתבצעות בדיקות יחידה (Unit tests) כחלק בלתי נפרד מתת שלב העיצוב והקידוד.

בדיקת המערכת מתבצעת באופן שוטף לכל אורך תהליך הפיתוח. סוגי בדיקות שונים מתבצעים על ידי גורמים שונים, בשיטות וכלים שונים כמעט בכל שלב או תת-שלב של מחזור החיים של הפרויקט. המטרות הנקודתיות הן שונות, אבל המטרה הכללית היא אחת: לאמת ולתקף את מהימנות ונכונות המערכת. ומכאן השם: אימות והוכחת תקפות, V&V (Verification & Validation), או: תורת הבדיקות.

מטרת מדריך זה היא "לעשות סדר" ולהתוות את התמונה הכוללת של בדיקות לאורך מחזור החיים וכיצד יש לבצע כל סוג של בדיקות. מקיט זה יש הפניות לקיטים אחרים העוסקים הלכה למעשה בבדיקות המתאימות, כגון: אפיון (הכנה ראשונה של תכנית הבדיקה), עיצוב ובנייה (בדיקות יחידה), שלב הבדיקות (בדיקות מערכת ובדיקות קבלה), התקנה והרצה (בדיקות מוכנות והתקנה) ועוד. קיט זה הוא "תורת" ו"מניפתי" ומטרתו לשמש מבוא ומתן ראייה כוללת לנושא יחד עם הפנייה מודרכת (אינדקס) שתאפשר לקורא לדעת לאן לגשת ומתי.

עקרונות מפת"ח

התמיכה הכוללת של מפת"ח באימות מערכת והוכחת תקפות (בדיקות לאורך מחזור החיים) מבוססת על מספר עקרונות:

- 1. מיקוד יחד עם כוללניות.** מחד גיסא, בדיקות חייבות להיעשות ב"ריכוז מאמץ" ומיקוד בנקודה מסוימת במחזור חיי המערכת (שלב הבדיקות) ובדגש על רכיבי המערכת הקריטיים והחיוניים ביותר. אי אפשר לבדוק הכל, כל הזמן ובכל המצבים. מאידך גיסא, בדיקות נכונות מלוות את תהליך הפיתוח לכל אורכו, מנסות לגלות תקלות עד כמה שיותר מוקדם ולאחר דווקא את אותם רכיבים ומצבים שנראים "פשוטים ותמימים". לעיקרון זה יש משמעות מעשית בחלוקת המשאבים המוקדשים לבדיקות ולאזונום בין החלק המוקדש ל"בדיקות מלוות פיתוח" ובין החלק המוקדש לשלב הבדיקות. המשאבים (התקציב) המיועדים לבדיקת המערכת חייבים להיות מחולקים נכון בין שני מוקדים אלה.
- 2. הבחנה בין גדלי פרויקטים.** בדומה לנושאים אחרים (כמו אפיון המערכת), גם באימות והוכחת תקפות יש להבחין בין גודל והיקף מערכת שונים. אין זה הגיוני להחיל על מערכת קטנה (היקף ג1 למשל) את אותן בדיקות של מערכת גדולה (ג3). הבחנה זו שזורה לאורך מפת"ח ומוסברת בקיטים שונים, בפרט מחזור החיים. (ארגונים הניגשים להסמכה ל- ISO 9000 או לתקני איכות אחרים עשויים אף להידרש שלא לקיים אבחנה זו, לחומרה).
- 3. תפיסת המטריצה.** כל סוג בדיקה מותאם לשלב מסוים במחזור החיים ובודק רכיב או רכיבים מסוימים בעץ המערכת. הקפדה על מודל המטריצה מאפשרת לאזן בין הבדיקות השונות, למנוע יתירות (Redundancy), לשמר המשכיות (Continuity) ועקיבות (Traceability) של הבדיקות לאורך מחזור החיים.

הקפדה על מודל המטריצה היא גורם מרכזי בתהליך המתמשך של אימות והוכחת תקפות לאורך מחזור חיי הפרויקט.

למה בודקים

מדוע מבצעים בדיקות לאורך מחזור חיי פרויקט:

- המטרה המרכזית היא לצאת לשוק עם מוצר אמין שיחזיר את ההשקעה.
- מניעת מצבים בהם המערכת אינה עובדת בצורה תקינה – בניית המוצר הלא נכון.
- מניעת מצב בו המערכת אינה עונה לדרישות שהוגדרו - בניית המערכת הלא נכונה.
- טעויות קורות תמיד, גם לצוותי פיתוח מיומנים. מטרת הבדיקות לאתר בזמן את הטעויות במטרה לתקן.
- מערכות תוכנה הן מערכות מורכבות הן ברמה הפונקציונלית והן ברמה הטכנולוגית. שילוב רמת המורכבות גורר סיבוכיות רבה הדורשת בדיקה מקיפה.
- ככל שנמצא שגיאה מוקדם יותר בפיתוח תוכנה כך עלות תיקונה תהיה קטנה יותר.



השלב במחזור החיים בו התגלתה השגיאה

איור מס' 1: עלות תיקון שגיאות בפיתוח תוכנה

סוגי בדיקות

"ארסנל" הבדיקות שעומד לרשות הבודק ומסייע לאימות המערכת והוכחת תקפותה מכיל את סוגי הבדיקות הבאים:

- Unit tests – בדיקות יחידה
 - Integration tests – בדיקות שילוב
 - System tests – בדיקות מערכת
 - Subsystem tests – בדיקות תת מערכת
 - Acceptance tests – בדיקות קבלה
 - Readiness tests – בדיקות מוכנות
 - Install tests – בדיקות התקנה
 - Regression tests – בדיקות רגרסיה
- להלן תיאור תמציתי של סוגי בדיקות אלה.

בדיקות יחידה – UNIT TESTS

בדיקות יחידה הן הרמה הבסיסית של בדיקות בהן נבדק כל מודול באופן נפרד כחלק בלתי נפרד מתהליך הקידוד (התכנות). הבדיקות מנוהלות באופן שוטף עבור כל מודול (תוכנית) בנפרד ומקושרות לתיק התכנות הרלוונטי. בדיקות היחידה מבוצעות לאחר ביצוע debugging ע"י התוכניתן – בדיקת התוכנית תוך כדי כתיבה בצורה אינטואיטיבית. ההנחיות ונתוני הבדיקה ניתנים על ידי המעצב והבדיקות עצמן נעשות על ידי התוכניתן כחלק מעבודת הקידוד השוטפת.

בדיקות יחידה מורכבות ממספר שלבים/דגשים:

- בדיקה מבנית – בדיקת רמת הסיבוכיות של הקוד, רמות הקינון, וכו'. במסגרת בדיקה זו ניתן לבצע בדיקות מעמיקות יותר הנקראות Code inspection בצורה ידנית או על ידי כלים המותאמים לשפת הפיתוח.
 - נתוני בדיקה – יש להכין נתוני בדיקה לקראת ביצוע הבדיקה הכוללים מעבר לנתונים סטנדרטים גם מקרי קצה.
 - הכנת מקרי בדיקה – יש לדעת מהן התוצאות הצפויות בעת הבדיקה ולהשוות אליהן את תוצאות הבדיקה בפועל.
 - ניסוי בפועל – ביצוע הבדיקות עצמן.
 - תיקון ובדיקה חוזרת – לאחר מציאת השגיאות יש לתקן ולבצע בדיקה שהשגיאה תוקנה.
- ביצוע הבדיקות מתועד בצורה מסודרת על גבי טפסים מתאימים, אם נדרש או בצורה טבלאית בה ניתן לראות כי כל אחת מהיחידות עברה בדיקות יחידה ונמצאה תקינה לאחר סיום הבדיקות, איתור התקלות וביצוע הבדיקות החוזרות.
- מנהל הפרויקט או מנהל צוות הפיתוח מאשר בסופו של דבר את ביצוע בדיקות אלה.
- בדיקות יחידה מבוצעות גם בשלב התחזוקה או בעצם בכל מצב בו מתבצע פיתוח של יחידות תוכנה.
- להרחבה נוספת ראה הקיט עיצוב ובנייה, נושא בדיקות יחידה, בכרך יסודות/מחזור חיים.

בדיקות שילוב – INTEGRATION TESTS

בדיקות שילוב הן בדיקות של מספר מודולים (יחידות תוכנה) ביחד במטרה לבדוק את תקינות הפעלתם יחדיו. כדי להכין את בדיקות האינטגרציה יש להכין רשימת מצאי ליחידות התוכנה הנבדקות: תוכניות, מסכים, קבצים וכו'.

הבדיקות מתבצעות בסביבת הניסוי אשר אמורה להיות מוכנה קודם לכן.

להרחבה נוספת ראה הקיט עיצוב ובנייה, נושא בדיקות שילוב, בכרך יסודות ומחזור חיים.

בדיקות מערכת – SYSTEM TESTS

שלב **בדיקות המערכת (System Test)** כולל מאפייני תכנון ובצוע כאחד. במסגרת שלב זה המערכת המפותחת נבדקת בשיטתיות וביסודיות במטרה לוודא שהיא **תקינה** - עומדת בדרישות שהוגדרו ע"י המשתמש בתחילת מחזור החיים ופועלת בהתאם להגדרות המפורטות בתיק האפיון.

בדיקות המערכת מבוצעות, בד"כ, על ידי צוות בדיקות מיוחד שאינו חלק מצוות הפיתוח. זאת במטרה לקבל נקודת מבט שונה על המערכת ולבצע בדיקות בצורה אובייקטיבית.

בדיקות המערכת מתועדות בתוכנית בדיקות מפורטת הכוללת:

- פירוט המשאבים (כ"א לבדיקות, ציוד נדרש, כלי בדיקה אוטומטיים וכו')
 - הכנת סביבת בדיקות, התואמת ככל האפשר את סביבת הייצור
 - הכנת תיק בדיקות הכולל מפרטי בדיקה (תרחישים ותסריטים) לכל רכיבי היישום הנבדקים (מסכים, תפריטים, תהליכים, פונקציות בתוכנה, וכו').
- בדיקות מערכת מורחבות כוללות גם נושאים רוחביים כגון: אבטחת מידע, בדיקות עומסים, בדיקות זמני תגובה ועוד.
- בדיקות המערכת מחולקות לסבבים ומחזורים. לכל סבב קיים מסמך המסכם את הממצאים. ממצאי הבדיקות עוברים בחזרה לצוות הפיתוח לתיקון ולבדיקות יחידה ואינטגרציה על פי הנדרש. לאחר קבלת מערכת מתוקנת מתבצע סבב נוסף.

בדיקות המערכת מסתיימות בעמידה במדדים וקריטריונים שהוגדרו מראש:

- כאשר לא מוצאים שגיאות נוספות
- כאשר הסתיים הזמן שהוקצב לבדיקות
- כאשר כמות השגיאות שנמצאו קטנה מערך שהוחלט עליו
- ועוד

בסיום שלב זה ניתן להחליט האם המערכת תעבור לשלב הבא: בדיקות הקבלה או שיש להחזיר את המערכת לצוות הפיתוח. במקרים חריגים אף ניתן להחליט להפסיק את הפרויקט בשלב זה.

להרחבה נוספת, ראה הקיט בדיקות מערכת בכרך יסודות ומחזור חיים.

בדיקות קבלה – ACCEPTANCE TESTS

בדיקות קבלה מתבצעות על ידי הלקוח לאחר סיום בדיקות המערכת. הבדיקות מתבצעות על פי תוכנית מוגדרת מראש בין הלקוח (מומחה היישום) לבין מנהל הפרויקט. תכנון הבדיקות וביצוען הם באחריות הלקוח. בדומה לצוות הפיתוח, גם אצל המשתמש עולה שאלה דומה לגבי זהות הצוות הבודק – האם יהיו אלה משתמשים שסייעו בהגדרת המערכת, השתתפו בסקרים ומכירים את

המערכת, כולל בעיות שנוצרו (שהם חלק מהן), או "משתמשי אמת" שייבחרו ע"י מומה היישום ואשר לא היה להם כל קשר לפיתוח המערכת. משתמשים אלה יכולים לספק זווית ראייה חדשה על המערכת, אך גם להעלות דרישות חדשות ובכך לסבך את הבדיקות. אפשר, כמובן לשלב שני צוותי בדיקה של משתמשים אחד מכל סוג.

עבור ביצוע בדיקות הקבלה מוקמת סביבה מיוחדת התואמת במירב האפשרי לסביבת הייצור.

הלקוח כותב תיק בדיקות קבלה ובו מפרטי בדיקות לכל חלקי המערכת הנבדקת.

ניתן לחלק את ביצוע הבדיקות לצוותי משנה של משתמשים המתמחים בתחומים שונים בהם מטפל היישום. בכל מקרה קיים אחראי אחד מטעם הלקוח לריכוז תוכנית בדיקות הקבלה, פיקוח על ביצוען ומתן הממצאים בצורה מסודרת בתיק בדיקות קבלה.

נקודת המבט בביצוע בדיקות הקבלה שונה מזו שמנחה את צוות בדיקות המערכת. הלקוח מוודא את תקינות המערכת בעיקר מבחינה תהליכית ומחקה את צורת עבודתו האמיתית, בעוד בדיקות המערכת מבוצעות באופן סדרתי כפי שאנשי מחשב מבינים את מודולריות המערכת.

בדיקות הקבלה יכולות להתבצע אל מול המערכת החדשה אך במקרים בהם המערכת החדשה מחליפה מערכת קודמת ניתן לבצע ריצה במקביל על שתי המערכות במטרה למצוא הבדלים בתגובת המערכות לתרחישים ולתסריטים המופעלים.

בסיום שלב זה הלקוח אמור לאשר את כניסת המערכת לעבודה בייצור.

להרחבה נוספת, ראה הקיט בדיקות מערכת בכרך יסודות ומחזור חיים.

בדיקות מוכנות – READINESS TESTS

בדיקות מוכנות משלימות את כל מה שלא נבדק בבדיקות המערכת ובדיקות קבלה. במקרים בהם בדיקות המערכת והקבלה התמקדו בבדיקת התוכנה, יש צורך בבדיקה משלימה של כל אותם רכיבים בעץ המערכת, בפרט רכיבים הנחוצים לתפעול השוטף ולתחזוקה, על מנת לוודא שהמערכת מוכנה להתקנה והעברה לייצור.

בפועל, בדיקת מוכנות איננה "ישות עצמאית" והיא מתבצעת באחד משני האופנים הבאים:

- במסגרת בדיקות המערכת (כאשר הן אינן מצטמצמות לבדיקות תוכנה בלבד ומקיפות רכיבים רבים של עץ המערכת)
- במסגרת בדיקות התקנה כמתואר בסמוך.

בדיקות תת מערכת – SUBSYSTEM TESTS

במערכות גדולות במיוחד ו/או במערכות בהן יש הקפדה חמורה ביותר על ניהול תצורה, מקובל גם לערוך בדיקות למכלולים או תת מערכות (Subsystem tests). בדיקות אלה הן מיזוג של בדיקות שילוב הנ"ל ובדיקות מערכת המתוארת בסמוך.

בפועל, בדיקות תת מערכת אינן "ישות עצמאית" והן יבוצעו באחד מהאופנים הבאים:

- במסגרת בדיקות השילוב, כחלק משילוב המערכת ובנייתה (מערכות קטנות ובינוניות)
- במסגרת שלב בדיקות המערכת (מערכות גדולות).

בדיקות התקנה – INSTALL TESTS

בדיקות התקנה מבוצעות בצורה מקיפה בעת התקנת המערכת בסביבת הייצור. בדיקות אלה נערכות גם בכל פעם שמתבצעת התקנה של המערכת בסביבה חדשה.

הבדיקות כוללות בדיקה של תכולת הגרסה המותקנת, בדיקת החומרה הרלוונטית עליה תותקן המערכת והתאמתה לנדרש וכן, בדיקות לאחר התקנה שהמערכת עובדת כנדרש. בדיקות תקינות המערכת מבוצעות באופן מדגמי אך במטרה לכסות את מרבית רכיבי המערכת: תפריטים, עבודה עם קבצים, דוחות, אבטחת מידע וכו'.

להרחבה נוספת ראה הקיט התקנה והרצה בכרך יסודות\מחזור חיים.

בדיקות רגרסיה – REGRESSION TESTS

בדיקות רגרסיה מבוצעות בכל פעם שמתבצעת התקנה של גרסה חדשה או שינוי כלשהו בסביבת העבודה של המערכת.

בעת פיתוח גרסה חדשה מבוצעים מספר סוגים של בדיקות:

- בדיקות יחידה ליחידות התוכנה שפותחו או שונו
- בדיקות אינטגרציה לבדיקת מספר יחידות תוכנה
- בדיקות מערכת חלקיות הבודקות את תקינות הרכיבים החדשים אשר פותחו במסגרת הגרסה או אלה ששונו.

שילוב שינויי הגרסה יכול לגרום בעיות ותקלות במערכת כולה ועל כן מבוצעות בדיקות רוחביות אשר מטרתן לוודא כי לא נפגעה תקינות חלקים אחרים של המערכת עקב הכנסת שינויי הגרסה.

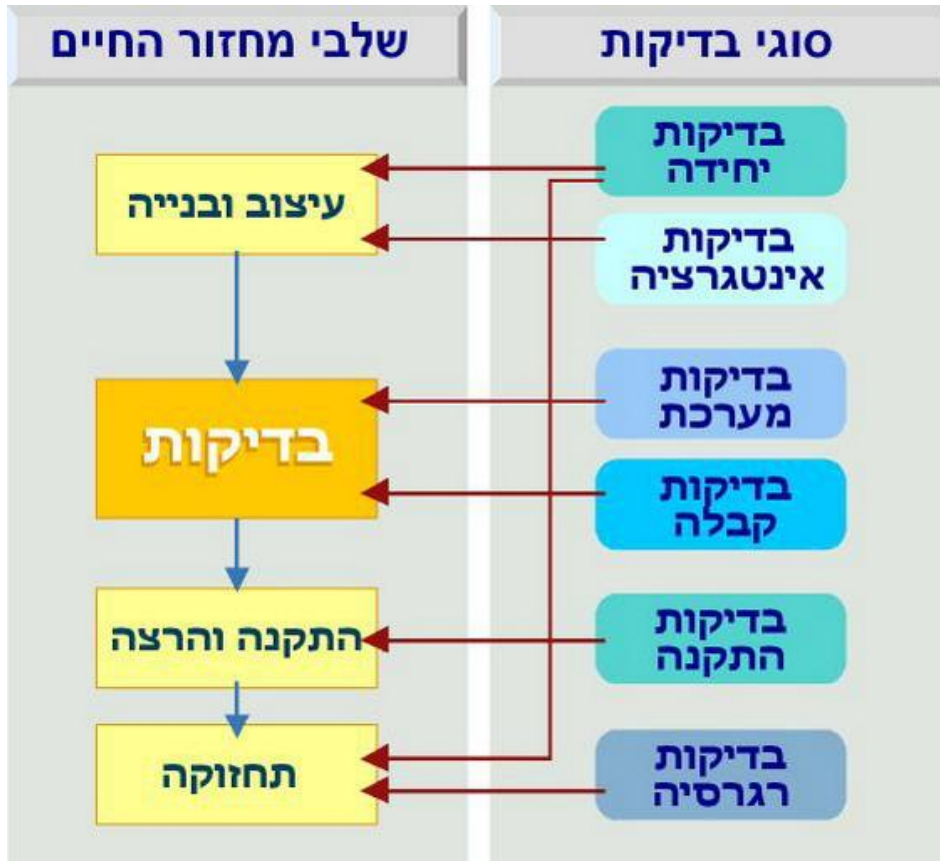
בדיקות אלה מתוכננות מראש באמצעות טפסי בדיקות. התסריטים שנכתבים אמורים לכסות בצורה מדגמית את כלל המערכת. בכל פעם שמכניסים גרסה חדשה יש לוודא כי התסריטים שהוכנו מראש אכן מספקים ויש לתקנם לאור הממצאים כך שיכילו הבדיקות הרצויות.

להרחבה נוספת ראה הקיט תחזוקה בכרך יסודות\מחזור חיים.

מימוש סוגי הבדיקות

את סוגי הבדיקות המפורטים בסעיף הקודם ניתן לבצע בכל נקודת זמן לאורך מחזור חיי הפרויקט, בהתאם לנדרש. בכל ארגון יסתייעו בסוגי בדיקות אלה וישבצו אותם בתהליך הפיתוח בהתאם לנסיבות וצרכי הפרויקט הנדון.

עם זאת, ככלל, השיבוץ המומלץ של סוגי הבדיקות בשלבי מחזור החיים של הפרויקט הוא כמוצג באיור מס' 2.



איור מס' 2: סוגי בדיקות במחזור החיים

שיבוץ במחזור החיים

מנקודת המבט של שלבי מחזור החיים אל סוגי הבדיקות (היפוך התמונה המוצגת באיור מס' 2 לעיל), נקבל את התמונה המצטיירת בטבלה הבאה:

היכן במפת"ח	סוג הבדיקות	שלב במחזור החיים
קיט עיצוב ובנייה בכרך יסודות/מחזור חיים.	בדיקות יחידה בדיקות שילוב בדיקות לתת מערכת ¹	עיצוב ובנייה
קיט בדיקות מערכת בכרך יסודות/מחזור חיים.	בדיקות מערכת בדיקות קבלה בדיקות לתת מערכת ¹ בדיקות מוכנות ²	בדיקות

היכן במפת"ח	סוג הבדיקות	שלב במחזור החיים
קיט התקנה והרצה בכרך יסודות\מחזור חיים.	בדיקות התקנה בדיקות מוכנות ²	התקנה והרצה
קיט עיצוב ובנייה בכרך יסודות\מחזור חיים.	בדיקות יחידה בדיקות רגרסיה	תחזוקה

טבלה מס' 1: שיבוץ סוגי הבדיקות בשלבי מחזור החיים של הפרויקט

מקרא:

¹ ניסוי לתת-מערכת מבוצע בשלב עיצוב ובנייה (תת-שלב שילוב) במערכות קטנות, או בשלב הבדיקות במערכות בינוניות וגדולות.

² בדיקות מוכנות יבוצעו בשלב בדיקות המערכת או בשלב התקנה והרצה.

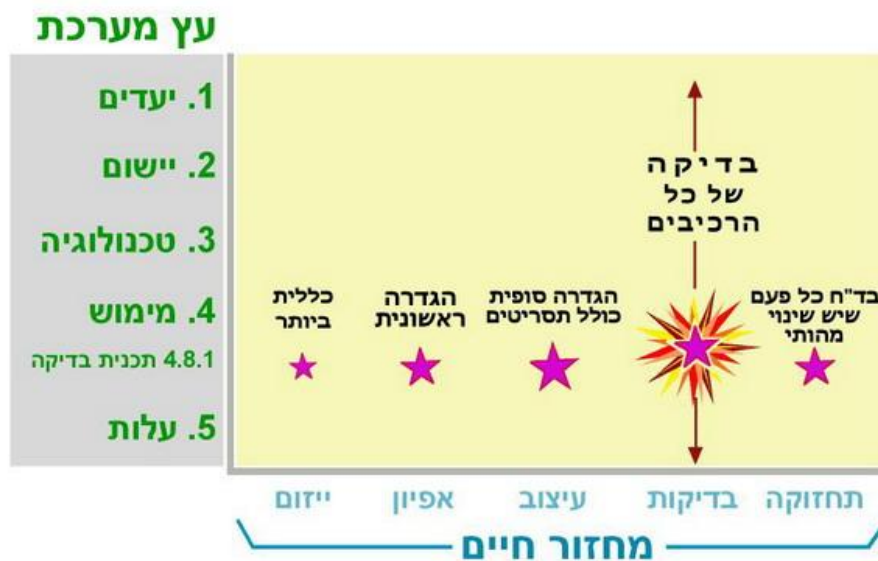
כוללניות מול מיקוד

שיבוץ זה של הבדיקות בשלבי מחזור החיים של הפרויקט ממחיש באופן ברור את המהות של פעולת האימות והוכחת התקפות כחוצה את תהליך הפיתוח הכולל של הפרויקט ואת עקרון הכוללניות \ מיקוד של הבדיקות שהוזכר לעיל בסעיף מבוא - הסבר כללי. סוגי הבדיקות ניתנים למימוש והפעלה בכל נקודת זמן ועם זאת, יש המלצה ברורה איזה סוג בדיקה יופעל באיזה שלב. נושא זה מוסבר בהרחבה בסעיף לאורך מחזור החיים בהמשך המדריך. אך לפני כן, נתמקד בשלב הבדיקות שהוא בודאי הציר המרכזי של בדיקות המערכת ונראה את עקרון המיקוד\כוללניות המאפיין את פעולת האימות והוכחת תקפות גם שם.

שלב הבדיקות

אחרי כל סוגי הבדיקות למיניהם, שלב הבדיקות (System tests) הכולל גם את מרבית בדיקות הקבלה של המשתמש (User acceptance tests), היה ונשאר הציר המרכזי של בדיקות המערכת. אין תחליף לבדיקה וניסוי מקיפים וכוללים של המערכת אשר מדמים את התנהגותה בתפעול ובייצור השוטף, כולל עמידה בעומסים, שמירה על שלימות (בסיס) הנתונים, יציבות והתאוששות, תפעול בחירום ומצבים חריגים וכו'. שלב הבדיקות הוא התחנה במחזור חיי הפרויקט בה צריכים להחליט אם משחררים את המערכת לתפעול וייצור אם לאו: Go/No-Go.

אלא שגם שלב הבדיקות אינו מצטמצם "לשלב עצמו" ומתפרס כמעט על מחזור החיים כולו כמוצג ב איור מס' 3 ובאיור מס' 4 להלן.



איור מס' 3: התפתחות תכנית הבדיקה לאורך מחזור החיים של הפרויקט

פריסת שלב הבדיקות מתבטאת בפעולות של "לפני" ו"אחרי" בדיקות המערכת:

- "לפני" - תכנון הבדיקות
- "אחרי" - שימוש חוזר

תכנון הבדיקות

התייחסות לתכנון הבדיקות יכולה להתחיל כבר בשלב הייזום. היא הכרחית בשלב האפיון בו מופיעה תוכנית בדיקה ראשונית. סעיף 4.8.1 בתיק האפיון מכיל את תמצית מפרטי הבדיקות והערכה כללית של תכנית הבדיקה: משאבים, סוגי בדיקות נדרשות, עקרונות לביצוע הבדיקות, שימוש בכלים, סביבות עבודה וכו'. בהמשך ההתקדמות במחזור החיים תתעדכן תוכנית הבדיקות הכללית ותיכתב באופן מפורט בפרק המנהלה של תיק בדיקות המערכת. בכל מקרה, בין אם המאפיין והמעצב בנו תכנית בדיקה, בין אם לאו, הגורם הבדוק יבנה תיק בדיקות מלא ומפורט לכל אותם רכיבים בעץ המערכת שנדרש לבדוק אותם.

שימוש חוזר

בתכנית הבדיקה (תיק הבדיקות) הושקעו מאמצים רבים ויש בה ערך רב להמשך הפרויקט: לשלב התחזוקה ולפיתוחים נוספים (מהדורות ויחידות מסירה שיתווספו למערכת בהמשך). יש

לעשות כל מאמץ לשמר את תכנית הבדיקה ואת תיק הבדיקות עבור שימוש חוזר: בדיקות רגרסיה ובדיקות חוזרות חלקיות שיידרשו בכל נקודת זמן. רכיב 4.8.1 של תיק המערכת (תיק התחזוקה) יצביע על תכנית הבדיקה המלווה את המערכת. תכנית זו היא "הצב"ד" (ציוד בדיקה) של המערכת.

איור מס' 4 מציג את פריסת שלב בדיקות המערכת במבט דומה אשר מדגיש את הפעילויות והתוצרים של התפתחות תוכנית הבדיקה לאורך כל מחזור החיים:



איור מס' 4: פריסת שלב הבדיקות על פני מחזור החיים של הפרויקט

לאורך מחזור החיים

הטבלאות הבאות מסכמות את נושא אימות והוכחת תקפות ופריסתו על פני מחזור החיים של הפרויקט.

כל שלבי מחזור החיים

הטבלה הבאה מתמקדת בשלבי מחזור החיים, בהם יש נגיעה ישירה, או עקיפה, לנושא אימות המערכת, בדיקתה והוכחת תקפותה.

התייחסות לאימות והוכחת תקפות (V&V)	שלב במחזור החיים
אזכור כללי ביותר ברכיב 4.8 חוסן ואמינות.	ייזום
רכיב 4.8.1 בתיק האפיון : תכנית בדיקה ראשונית.	אפיון
בדיקות יחידה Unit Tests בדיקות שילוב Integration Tests ¹ המשך "עיבוי" ועיצוב תכנית הבדיקה (רכיב 4.8.1).	עיצוב ובנייה
בניית תכנית הבדיקה תיק בדיקות מערכת תיק סיכום בדיקות בדיקות קבלה ע"י המשתמשים ומומחה היישום הנחיות ניהוליות ומקצועיות שונות (לצוותים, סבבים ומחזוריים).	בדיקות מערכת System Tests -
בדיקות התקנה [בדיקות מוכנות]	התקנה והרצה
בדיקות יחידה בדיקות גרסיה	תחזוקה
בדיקות גרסיה בדיקות חוזרות	יחידות מסירה ופיתוחים נוספים

טבלה מס' 2: ההתייחסות לאימות והוכחת תקפות לאורך מחזור החיים של הפרויקט

תוצרי בדיקות מול תוצרי המערכת

הטבלה הבאה מסכמת את נושא אימות והוכחת תקפות במבט של רשימת תוצרי הבדיקות השונים וייחוסם לתוצרי המערכת עצמה, כפונקציה של שלבי מחזור החיים והיקף הפרויקט:

תוצרי בדיקות	תוצר תיעוד ראשי	השלב
4.8 אזכור כללי לעצם הצורך, היקף ועומק הבדיקות הנדרשות.	מסמך ייזום	ייזום
4.8.1 תכנית בדיקה תמציתית (על בסיס נספח 4.8.1 הבסיסי).	מסמך אפיון	אפיון ג1
4.8.1 מורחב על בסיס תיק בדיקות פורמט 1/2 או נספח 4.8.1	תיק אפיון	אפיון ג2
4.8.1 מצביע על תיק בדיקות לפרויקט (TRD מלא) אשר נבנה במקביל לתיק האפיון ועל בסיס גלופת תיק הבדיקות, בהתאמה לשלב האפיון (דרישות לטסטים)	תיק אפיון	אפיון ג3
הפיכת TRD ל- STD ע"י הוספת פרוצדורות בדיקה, נתוני בדיקה סופיים, תסריטים ועוד. STD הוא בהתאם לגודל	תיק עיצוב	עיצוב

השלב	תוצר תיעוד ראשי	תוצרי בדיקות
		הפרויקט : ג1 - 4.8.1 בסיסי ג2 - 4.8.1 מורחב (תיק בדיקות מקוצר פורמט 1/2) ג3 - תיק בדיקות מלא פורמט 1/2
בנייה	המערכת עצמה תיק עיצוב מאושר	בדיקות יחידה (unit test) מתועדות בתיקי התכנות
בדיקות מערכת	המערכת עצמה	תיק סיכום בדיקות ו-STD משופר, בהתאם להיקף המערכת : ג1, ג2, ג3, כני"ל בשלב העיצוב. בדיקות קבלה ע"י המשתמש
התקנה והרצה	המערכת עצמה תיק מערכת	טופס אישור לבדיקות התקנה.
תחזוקה	ניהול שינויים וגרסאות	בדיקות יחידה (unit test) מתועדים בתיקי התכנות. עדכונים שוטפים של תסריטי הבדיקות או תיק הבדיקות לפי גודל המערכת. תסריטים הנכללים בבדיקות הרגרסיה.
פיתוחים ויחידות מסירה נוספות	תיקי נושא עדכונים לתיק המערכת	בדיקות יחידה (unit test) מתועדים בתיקי התכנות. תיקי בדיקות לנושאים (יחידות מסירה) [עדכון תיק הבדיקות הכולל] תסריטים הנכללים בבדיקות הרגרסיה.

טבלה מס' 3: תוצרי המערכת ותוצרי הבדיקות (אימות והוכחת תקפות) לאורך מחזור החיים של הפרויקט

עץ המערכת

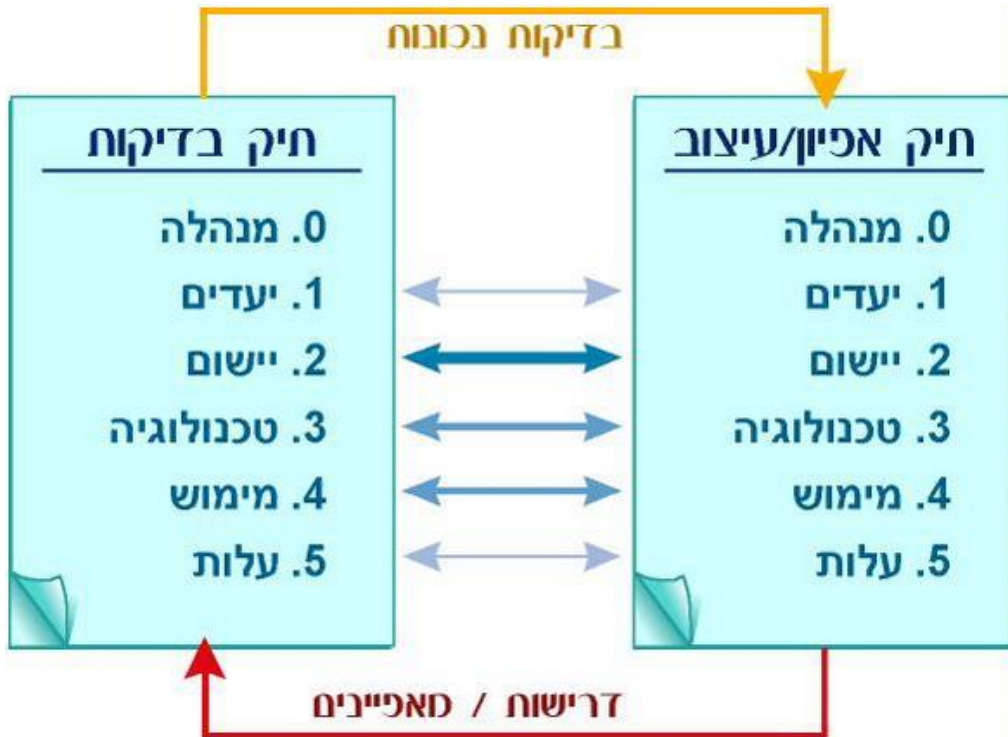
בעץ המערכת יש רכיב אחד המוקדש לבדיקת המערכת: רכיב 4.8.1 תכנית בדיקה. רכיב זה מופיע באופן עקבי בכל תיקי המערכת, מתפתח לאורך הפרויקט (ראה איור מס' 3 ומותאם להיקף המערכת כמוסבר להלן). בהתאם לגודל המערכת ולשלב בו נמצא הפרויקט, יהיה רכיב 4.8.1 באחד מהאופנים הבאים:

- 3-1 עמודים בגוף התיק (תיק אפיון) אשר מתאר בקצרה תכנית בדיקה ראשונית – מתאים למערכות בינוניות ומטה
- נספח 4.8.1 אשר מגדיר תכנית בדיקה בסיסית, כולל מנהלה ראשונית ומפרטי בדיקה עיקריים. מבנה זה מתאים לתיק אפיון למערכות גדולות ובתיק העיצוב למערכות בינוניות ומטה.
- מסמך נפרד "תיק בדיקות תכנית בדיקה", המוצב מרכיב 4.8.1. מבנה זה מתאים לתיק אפיון למערכות גדולות מאד, בתיק העיצוב למערכות בינוניות ומעלה ובשלב הבדיקות לכל סוגי המערכות בכל הגדלים וההיקפים.

רכיב נוסף הקשור בנושא בדיקת המערכת הוא רכיב 3.13 אשר מגדיר את כלי הפיתוח, כולל כלים ייעודיים לבדיקות (ראה סעיף 6 בהרחבה לרכיב 3.13 בעץ המערכת האוניברסלי).

עם זאת, כל רכיבי עץ המערכת מעורבים בבדיקת המערכת, אימותה והוכחת תקפותה. כל רכיב המוגדר באופן ברור, ניתן לכימות וכו', תורם לבדיקות (Testability) של המערכת. ולחילופין, כל רכיב שאינו מוגדר באופן ברור, מקשה על אימות המערכת והוכחת תקפותה. היוצא מכך, שעץ המערכת כולו, לא רק רכיבים 4.8.1 ו-3.13 קשורים לבדיקות, אימות והוכחת תקפות. רעיון זה מוצג באיור מס' 5. מטרת תיק הבדיקות (שלב הבדיקות כולו) היא לאמת את הדרישות שהוגדרו באפיון ועוצבו בשלב העיצוב ובנייה. תפקיד תיק האפיון והעיצוב הוא להזין את תיק הבדיקות במידע הדרוש על מנת להגדיר את התרחישים, התסריטים, נתוני הבדיקה והתוצאות המצופות. ללא הגדרה כזו ייאלץ צוות הבדיקות להגדיר בעצמו את דרישות המערכת – מול מה בודקים – היינו, לבנות תיק אפיון/עיצוב "לאחור".

זכור, באג מוגדר כאי תאימות התיעוד למציאות.



איור מס' 5: תיק הבדיקות הוא "הצל" של תיק האפיון

איור מס' 5 מדגים את עקרון מס' 3: תפיסת המטריצה שהוצג בסעיף עקרונות מפת"ח במבוא לעיל.

תקן האיכות ISO9000 2000

לאימות והוכחת תקפות מוקדש סעיף 7.3.5 אימות תכן ופיתוח וסעיף 7.3.6 תיקוף תכן ופיתוח בתקן האיכות ISO 9000 2000. ארגונים הניגשים לבדיקה וקבלת אישור ל-ISO-9000 יתחילו משם.

ראה הקיט תקן ISO9000 בכרך איכות.

הרחבות

בדיקות במערכות קטנות

במסגרת מערכות קטנות מבוצעות בדיקות לכל אורך פיתוח המערכות. יש לבחון את רמת הבדיקות הנדרשת והחלוקה לסוגי הבדיקות כפי שהוצגו לעיל.

בדיקות יחידה מבוצעות בכל מקרה אך בהחלט ניתן לבצע בדיקות אינטגרציה תוך ביצוע בדיקות היחידה כיוון שהיקף המערכת נאפשר זאת.

בדיקות המערכת מבוצעות בצורה מלאה. אין צורך להכין תיק בדיקות מלא אלא ניתן להסתפק בתסריטי בדיקות.

להרחבה ראה קיט מערכות קטנות בכרך נושאים תומכים.

בדיקות בפיתוח RAD

(Rapid Application Development – RAD) בדיקות מבוצעות בצורה מקיפה גם כאשר צורת הפיתוח שונה ממחזור חיים סדרתי. כאשר מפתחים יחידת מסירה בכל פעם מבוצעות בדיקות היחידה והאינטגרציה באופן שוטף.

תיק בדיקות מערכת נכתב בפעם הראשונה במהלך פיתוח יחידת המסירה הראשונה. במסגרת תיק זה באים לביטוי היבטים רוחביים כגון: זמני תגובה, ממשק משתמש סטנדרטי, אבטחת מידע וכו' וכן כל הרכיבים שפותחו ביחידת מסירה זו. הבדיקות מבוצעות לפי המקובל בשלב הבדיקות. ייתכן כי חלוקה ליחידות מסירה תיצור יחידת מסירה קטנה התואמת מערכת קטנה. במקרה זה יש לפעול על פי ההתייחסות למערכות קטנות. עם זאת, יש לבחון את המערכת בראיה עתידית. המערכת העתידית עשויה להיות מערכת בינונית או גדולה ואז יש לבחון התנהגות בנושא הבדיקות בגישה הרגילה כבר ביחידת המסירה הראשונה.

בהמשך, עם שחרור יחידות מסירה נוספות, בדיקות היחידה יתבצעו באופן רגיל ובדיקות האינטגרציה יתבצעו תוך שילוב יחידת מסירה זו ביחידות המסירה הקודמות. לאחר מכן יבוצעו בדיקות מערכת אשר יכללו מצד אחד בדיקות מקיפות ליחידת המסירה הנוכחית אך יכללו גם בדיקות גרסיה כלפי תפקוד יחידות המסירה הקודמות, על מנת לוודא ששילוב יחידת מסירה זו לא פגע בתקינות המערכת הקיימת.

בדיקות קבלה יתבצעו על ידי המשתמש כמקובל.

להרחבה ראה הקיט מחזורי חיים דינמיים בכרך ניהול.

טכניקות וכלים

באימות והוכחת תקפות מקובל להיעזר בטכניקות הבאות:

- בדיקות קופסא שחורה – מה נכנס מה יוצא בלי בדיקה פנימית
- בדיקות קופסא לבנה (שקופה) - מה נכנס מה יוצא כולל בדיקה פנימית
- קבוצות שקולות
- ניתוח תחומים
- ניחוש שגיאות
- הכנת נתוני בדיקה
- מקרי קצה
- אכלוס נתונים

כל האמצעים המוצגים בקיט זה (ובקיטים המופנים ממנו) הם טכניקות, שיטות ונהלים שמפת"ח מציע. בנוסף, יש לזכור את הכלל המרכזי לפיו מפת"ח הוא נוהל מסגרת ויש תמיד מקום לבתי תוכנה ולחברות המתמחות בבדיקות להוסיף טכניקות מפורטות ואפילו מתודולוגיות אשר ישתלבו סביב מפת"ח ויסייעו בבדיקות המערכת. עדיף כמובן, שימוש בתוכנות וכלים ממוחשבים לבדיקות, להכנת נתוני בדיקה, לאכלוס קבצים, בדיקות יחידה וכו'. כל תוכנה, טכניקה ומתודולוגיה התורמים לבדיקות המערכת והגברת אמינותה ואיכותה - מקובלים. ובלבד שכלים אלה יפעלו במתואם עם מפת"ח ויפיקו את התיעוד הנדרש. תוספות אלה, כולל החיבוריות למפת"ח, אינן חלק רשמי ממפת"ח והן משוקות ע"י בתי התוכנה ובאחריותם.