**תבנית למפרט אפיון ותכן – פרויקט שנתי בהנדסת תוכנה**

תבנית זו משמשת כבסיס לכתיבת מפרט האפיון והתכן של הפרויקט. יש להעתיק ולשמור עותק פרטי של התבנית תחת השם XXX\_Project\_Spec&Design\_v1.docx, כאשר XXX הוא שם הפרויקט. לאחר מכן יש לערוך את המסמך על פי ההנחיות לשימוש בתבנית בהמשך.

במידה ובמהלך הפרויקט חלים שינויים במסמך, יש לייצר גרסה חדשה (v2, v3 בשם הקובץ וכו') ולפרט בקצרה את השינויים בטבלת מעקב הגרסאות שבתחילת המסמך. כמו כן יש לעדכן את מספר הגרסה בכותרת העמודים.

את מסמך האפיון והתכן יש להגיש בשני עותקים מודפסים (בצבע, אם רלוונטי).

הנחיות לשימוש בתבנית:

* כל טקסט המופיע בכיתוב שחור – **יישאר כפי שהוא**.
* כל טקסט המופיע <בצבע אדום בין צומדיים> **יוחלף בטקסט רלוונטי**, שייכתב בשחור (ללא הצומדיים)
* כל טקסט המופיע *בצבע כחול מוטה* מהווה הנחייה או דוגמה, ויש **להשמיטו מהמסמך**.
* כל קטע המופיע בתוך מסגרת כחולה מוצללת משמש כהנחיה ויש **להשמיטו מהמסמך בשלמותו**.
* **אין להשמיט** סעיפים מהמסמך:  
  - בסעיף שתוכנו לא ישים לפרויקט ייכתב "לא ישים"  
  - בסעיף שתוכנו טרם נקבע ייכתב "טרם נקבע" או TBD.
* יש לשמור על פורמט המסמך, ובעיקר על הסגנונות "רגיל", כותרות 1,2,3, "כיתוב" ו- TOC 1,2,3.
* יש לשמור על הכותרת העליונה והתחתונה, כולל מיספור העמודים בתחתיתם.
* כל איור יכיל כותרת ממוספרת מתחתיו, לדוגמה: **איור 5: תרשים זרימה**. מומלץ להשתמש בתכונת המיספור האוטומטי של איורים ב-word (שימוש בסגנון "כיתוב"). יש לוודא שהאיורים קריאים וברורים!
* כל טבלה תכיל כותרת ממוספרת מעליה, לדוגמה: **טבלה 2: משתתפי הפרויקט**. מומלץ להשתמש בתכונת המיספור האוטומטי של טבלאות ב- word (שימוש בסגנון "כיתוב").
* את תוכן העניינים של המסמך, רשימת האיורים, ורשימת הטבלאות יש לייצר בצורה אוטומטית.

תכנים שמקורם בקוד, בתוכנות אחרות או במסמכים חיצוניים ייכללו במסמך **באחת** מהצורות הבאות:

1. העתקה/הדבקה במקום הרלוונטי בגוף המסמך
2. הוספה כנספח בסוף המסמך (יש למספר ולהכניס לתוכן העניינים), תוך הפניה מהסעיף הרלוונטי לנספח
3. צירוף כקובץ נלווה למסמך (יש לכלול ברשימת הסימוכין שבסוף המסמך) תוך הפניה מהסעיף הרלוונטי.

פרויקט <שם הפרויקט>

שנה"ל <תשע"<X

מפרט אפיון ותכן

גרסה <מספר גרסה>

תאריך הגשה: <תאריך>

|  |
| --- |
| **מבצעי הפרויקט** |
| <שם הסטודנט>, ת.ז. <מס' ת.ז.>, <מחלקה/מסלול>  <שם הסטודנט>, ת.ז. <מס' ת.ז.>, <מחלקה/מסלול> |
| **הלקוח** |
| <שם החברה / הגוף / האדם עבורם מבוצע הפרויקט> |
| **הנחיה תעשייתית** |
| <מנחה תעשייתי>, <תפקיד>, <חברה>  <מנחה תעשייתי>, <תפקיד>, <חברה> |
| **הנחיה אקדמית** |
| <מנחה אקדמי>, <מחלקה>, <מוסד>  <מנחה אקדמי>, <מחלקה>, <מוסד> |

**מעקב גרסאות**

|  |  |  |
| --- | --- | --- |
| **גרסה** | **תאריך** | **מהות השינויים מהגרסה קודמת** |
| 1 |  | גרסת בסיס ראשונה |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

תוכן העניינים

*תוכן עניינים זה נוצר אוטומטית ויש לעדכנו בכל פעם שנעשים שינויים במסמך. העדכון מתבצע דרך תפריט "הפניות > עדכון תוכן עניינים".*

1 רקע 7

2 הסביבה הארגונית והתהליכים העסקיים 7

3 מילון מונחים 7

4 מפרט דרישות המערכת/התוכנה 7

4.1 בעלי עניין ושחקנים 7

4.2 מקורות הדרישות 8

4.3 טבלת דרישות מערכת/תוכנה 8

4.4 מפרט דרישות מערכת/תוכנה 9

4.4.1 <שם Use Case> יש להשתמש בדיוק באותו שם המופיע בתרשים! 9

5 מפרט ארכיטקטורה 10

5.1 ארכיטקטורה פיזית 10

5.2 ארכיטקטורה לוגית 11

5.2.1 ארכיטקטורה לוגית ל<פריט/צומת/תוכנת> <שם הפריט/הצומת/התוכנה> 11

5.3 ארכיטקטורה כוללת 11

6 מפרט תכן תוכנה 12

6.1 מודל מחלקות 12

6.1.1 תרשים מחלקות למארז/רכיב <שם המארז/הרכיב> 12

6.2 מודל התנהגות 12

6.2.1 תרשימי רצף ל-Use Case <שם ה-UC> 13

7 יישום ובניה (מדריך למתכנת) 13

8 מפרטי בדיקות 13

8.1 מפרט בדיקות <שם סבב הבדיקות, למשל קבלה> 13

8.1.1 התוצר הנבדק 13

8.1.2 סביבת הבדיקות 13

8.1.3 פירוט הבדיקות 13

9 התקנה, שימוש ותיקוף 14

9.1 התקנה 14

9.2 שימוש 14

9.3 תיקוף 14

10 סימוכין 14

רשימת איורים

*רשימת איורים זו נוצרה אוטומטית ויש לעדכנה בכל פעם שנעשים שינויים במסמך. העדכון מתבצע דרך תפריט "הפניות > עדכון טבלה".*

[איור 1: תרשים Use Cases של מערכת/תוכנת <שם המערכת/התוכנה> 9](#_Toc437179628)

[איור 2: הארכיטקטורה הפיזית של מערכת <שם המערכת> 10](#_Toc437179629)

[איור 3: הארכיטקטורה הלוגית (רכיבים וממשקים) של <פריט/צומת/תוכנת> <שם הפריט/הצומת/התוכנה> 11](#_Toc437179630)

[איור 4: הארכיטקטורה הכוללת של <פריט/צומת/תוכנת> <שם הפריט/הצומת/התוכנה> 12](#_Toc437179631)

[איור 5: תרשים מחלקות של מארז/רכיב <שם המארז/הרכיב> 12](#_Toc437179632)

[איור 6: תרשים רצף של <תרחיש/ים> 13](#_Toc437179633)

רשימת טבלאות

*רשימת טבלאות זו נוצרה אוטומטית ויש לעדכנה בכל פעם שנעשים שינויים במסמך. העדכון מתבצע דרך תפריט "הפניות > עדכון טבלה".*

[טבלה 1: בעלי עניין ושחקנים של מערכת/תוכנת <שם המערכת/התוכנה> 7](#_Toc437179623)

[טבלה 2: צמתים 10](#_Toc437179624)

[טבלה 3: פריטי תוכנה 10](#_Toc437179625)

[טבלה 4: ממשקי תקשורת (פיזיים) ופרוטוקולים 10](#_Toc437179626)

[טבלה 5: טבלת רכיבים וממשקים ל<פריט/צומת/תוכנת> <שם הפריט/הצומת/התוכנה> 11](#_Toc437179627)

# רקע

מסמך זה מהווה מפרט טכני <למערכת/לתוכנה> המפותחת ומסופקת במסגרת פרויקט <שם הפרויקט>, וכולל את הגדרת הדרישות והתכן ואת פירוט אופן הביצוע והבדיקה של הפרויקט, בכפוף ליעדים, למאפיינים ולתכנון שבסימוכין [1].

*טקסט, בהיקף של כחצי עמוד עד עמוד אחד, הסוקר* ***בקצרה*** *את מסמך הייזום ומפרט את היעדים העיקריים, הדרישות העיקריות, המאפיינים העיקריים, אבני הדרך העיקריות וכל מידע משמעותי שנוצר או התגבש בין כתיבת מסמך הייזום לבין כתיבת מסמך זה.*

# הסביבה הארגונית והתהליכים העסקיים

*פרק זה מפרט את הסביבה הארגונית בה משולבת המערכת/התוכנה המפותחת במסגרת הפרויקט. יש לפרט תהליכים ארגוניים/עסקיים, משתמשים, בעלי עניין, מערכות חיצוניות וכו'. ניתן להביא שוב את התרשים הסכמטי מתוך מסמך הייזום. לפירוט התהליכים הארגוניים מומלץ לערוך תרשים Use Cases ארגוני, אולם אין חובה לכתוב מפרט Use Case מלא לתהליכים אלה – די לפרטם בטקסט. במידה ורוצים להציג באופן גרפי את זרימת התהליכים (לוגיקה עסקית/ארגונית) יש להשתמש ב- Activity Diagrams של UML.*

# מילון מונחים

*רשימה או תרשים PDOM (Problem Domain Object Model) המגדירים את הישויות/המונחים העיקריים במרחב היישום (application domain) ואת הקשרים ביניהם.*

# מפרט דרישות המערכת/התוכנה

## בעלי עניין ושחקנים

בטבלה 1 להלן מפורטים בעלי עניין ושחקנים של המערכת/התוכנה.

**שחקן** הוא ישות כלשהי מחוץ למערכת (אנושית או ממוכנת) המבצעת אינטראקציה ישירה עם המערכת/התוכנה. אינטראקציה זו משקפת את **היעדים** אותם מעוניינים השחקנים להשיג באמצעות המערכת/התוכנה, או את השירותים אותם מבקשת המערכת/התוכנה לקבל מהם.

*לדוגמה: משתמשים, מפעילים, מערכות בדיקה, מערכות תחזוקה, יישומים אחרים, רכיבים חיצוניים וכו'.*

**בעל עניין** הוא ישות כלשהי (אדם, ארגון, גוף, בעל תפקיד וכו') ה**משפיעה** על פעולת המערכת או **המושפעת** מפעולתה, למרות שאינה מבצעת אינטראקציה ישירה איתה. מעורבות זו בפעולת המערכת משקפת את **האינטרסים** של בעלי העניין במערכת זו.

*לדוגמה: גופים ארגוניים, גופים עסקיים, גופי רגולציה/תקינה וכו'. שחקני המערכת הינם בעלי עניין מטבעם.*

טבלה 1: בעלי עניין ושחקנים של מערכת/תוכנת <שם המערכת/התוכנה>

|  |  |  |
| --- | --- | --- |
| **שם/מהות** | **שחקן?** | **אופן המעורבות בפעולת המערכת/התוכנה (יעדים/אינטרסים)** |
| *לדוגמה:משתמש* | *כן* | *מבצע פעולות בכספומט (משיכה, הפקדה וכו')* |
| *לדוגמה: מחשב מרכזי* | *כן* | *הכספומט משתמש בו לצורך איחזור ואיחסון מידע במהלך ביצוע פעולות* |
| *לדוגמה: קב"ט* | *לא* | *המידע העובר אל ומאת הכספומט - מאובטח* |

## מקורות הדרישות

דרישות המערכת/התוכנה המפותחת במסגרת פרויקט זה התקבלו מהמקורות הבאים:

*יש לערוך רשימה ממוספרת של מקורות* ***כתובים*** *מהם נלקחו הדרישות, ובראשם מסמך הייזום. את המקורות עצמם יש לפרט בסימוכין ולהפנות מכאן. מקורות נוספים לדרישות יכולים להיות תקנים רלוונטיים, מפרטי מערכת-על, מפרטים טכניים אחרים, סיכומי פגישות/ראיונות, הודעות דואר אלקטרוני ועוד. בעלי העניין הינם המקורות העיקריים של הדרישות!*

1. מסמך הייזום לפרויקט – סימוכין [1].

## טבלת דרישות מערכת/תוכנה

טבלת דרישות מערכת/תוכנה נערכה ב-MS-Excel ומצורפת למסמך זה כסימוכין [<מס'>]. הטבלה מציינת את מקורה של כל דרישה ואת סיווגה הפונקציונאלי ו/או הלא-פונקציונאלי. הדרישות התפעוליות בטבלה מצביעות ל- Use Cases רלוונטיים שבסעיף הבא, אשר אמורים לספק מענה לדרישות אלה.

*על בסיס מקורות הדרישות שהוזכרו, יש לבנות "טבלת דרישות מערכת/תוכנה" (בדרך כלל תכיל הטבלה מספר דרישות שהוא בערך פי 3-5 ממספר הדרישות שבמסמך הייזום). במידת הצורך ניתן לגזור דרישות מפורטות יותר מתוך הדרישות המקוריות, אולם יש להקפיד שלא לגלוש לרמה של מימוש או תכן מפורט. את הטבלה יש לערוך ב-Excel, על פי הפורמט המופיע בדוגמאות "הספריה הרובוטית" ו"חתרנית" שבאתר. כל שורה בטבלה תכלול:*

* *מספר דרישה*
* *נוסח הדרישה, צמוד ככל שאפשר לניסוח המקורי*
* *מקור הדרישה*
* *סיווג פונקציונאלי (דרישה תפעולית, דרישת מידע)*
* *סיווג לא-פונקציונאלי (ביצועים, מאפיין איכות, אילוץ חומרה, אילוץ מימוש, אילוץ ניהולי)*

*יש להקפיד שלכל דרישה יהיה לא יותר מסיווג פונקציונאלי אחד וסיווג לא פונקציונאלי אחד. במקרה כזה יש להפריד את הדרישה למספר דרישות.*

* *הצבעה של כל דרישה פונקציונאליות ל- Use Case מערכתי אחד או יותר.*

## מפרט דרישות מערכת/תוכנה

סעיף זה מכיל פירוט של דרישות המערכת/התוכנה בצורת Use Cases. תרשים Use Cases של כלל המערכת/התוכנה מופיע באיור <מס'> להלן. ה- Use Cases שבתרשים מפורטים בסעיפים הבאים.

Use Case Diagram

איור 1: תרשים Use Cases של מערכת/תוכנת <שם המערכת/התוכנה>

### <שם Use Case> יש להשתמש בדיוק באותו שם המופיע בתרשים!

|  |  |
| --- | --- |
| **שחקנים** | *יש לפרט את כל השחקנים הרלוונטיים ל-UC זה באופן הבא:*   * <שם השחקן>: שחקן ראשי *יש לפרט מה מנסה השחקן להשיג* * <שם השחקן>: שחקן תומך *יש לפרט מה המערכת צריכה ממנו* * Use Case ספונטאני *(אם ה-UC מופעל פנימית מתוך המערכת ואין לו שחקן ראשי)* |
| **בעלי עניין אחרים** | * <שם השחקן>: *יש לפרט את האינטרסים הרלוונטיים של בעל העניין ב-UC זה, כלומר מהי השפעתו על אופן פעולת ה-UC או כיצד הוא מושפע ממנה.* |
| **תנאים מקדימים**  **(pre-conditions)** | * *אחת או יותר הנחות שקיומם מהווה תנאי ליכולת ה-UC לפעול. לדוגמה: המערכת מקוונת.* |
| **תנאים לאחר-מעשה**  **(post-conditions)** | * *אחת או יותר תוצאות, אשר קיומן מהווה תנאי* ***לסיום מוצלח*** *של ה-UC (מנקודת הראות של השחקנים ובעלי העניין)* |
| **אירוע מפעיל**  **(trigger)** | *האירוע הגורם ל-UC להתחיל להתבצע. יכול להיות אחד (או יותר) מהבאים:*   * *פעולה של שחקן ראשי* * *אירוע פנימי במערכת* |
| **תרחיש הצלחה ראשי (MSS)** | *רשימה ממוספרת של צעדי האינטראקציה המתבצעת בין המערכת לשחקניה* ***לאחר*** *האירוע המפעיל. כל צעד מתחיל בשם של שחקן ראשי או ב"המערכת". לדוגמה:*   1. ***המערכת*** *מציגה לקורא טופס הרשמה* 2. ***הקורא*** *מזין פרטים* 3. ***המערכת*** *מוודאת שהפרטים תקינים ומלאים* 4. ***המערכת*** *רושמת את הקורא* 5. ***המערכת*** *מספקת לקורא סיסמה* |
| **הסתעפות א'** | *הסתעפויות הינן מסלולי אינטראקציה אלטרנטיביים. הסתעפויות המובילות לקיום התנאים לאחר-מעשה נקראות* ***חלופות****. הסתעפויות המובילות לסיום האינטראקציה ללא השגת התנאים לאחר-מעשה נקראות* ***חריגות****. יש לפרטן באופן הבא:*  <חלופה/חריגה> מצעד <מס'> <של MSS / של הסתעפות ...>: *תיאור הנסיבות שגרמו להסתעפות. לדוגמה: פרטי הקורא אינם מלאים.*  *תיאור האינטראקציה החלופית המתרחשת במהלך ההסתעפות, כולל ציון של חזרה לתרחיש הקודם (לאיזה צעד) או סיום התרחיש.* |
| **עקיבות לדרישות** | תפעוליות: *יש לציין את מספרי הדרישות בטבלת הדרישות המפנות ל-UC זה*  אחרות: *יש לציין דרישות אחרות (מידע, ל"פ) הרלוונטיות למימוש UC זה* |

# מפרט ארכיטקטורה

## ארכיטקטורה פיזית

הארכיטקטורה הפיזית של המערכת, המתוארת בתרשים <מס'> מפרטת את הישויות והקשרים הבאים: *התרשים ייערך כתרשים פריסה (Deployment Diagram).*

* **צמתים (Nodes)**: מרכיבי החומרה של המערכת
* **ממשקי תקשורת (Communication Links)**: הקשרים הפיזיים בין הצמתים והפרוטוקולים המשמשים לתקשורת
* **פריטי תוכנה (Software Artifacts)**: יישומים ומרכיבי תוכנה אחרים המותקנים בצמתי החומרה.

Deployment Diagram

איור 2: הארכיטקטורה הפיזית של מערכת <שם המערכת>

מרכיבי הארכיטקטורה הפיזית מפורטים בטבלאות הבאות:

טבלה 2: צמתים

|  |  |  |
| --- | --- | --- |
| **צומת** | **מהות** | **דרישות נתמכות** |
|  |  |  |
|  |  |  |

טבלה 3: פריטי תוכנה

|  |  |  |
| --- | --- | --- |
| **פריט** | **מהות** | **דרישות נתמכות** |
|  |  |  |
|  |  |  |

טבלה 4: ממשקי תקשורת (פיזיים) ופרוטוקולים

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **צומת** | **צומת** | **כיוון** | **תוכן** | **תווך / פרוטוקול** | **דרישות נתמכות** |
|  |  | <  >  <> |  |  |  |
|  |  |  |  |  |  |

## ארכיטקטורה לוגית

הארכיטקטורה הלוגית של המערכת, המפרטת את רכיבי התוכנה ואת ממשקיהם מתוארת בתרשימ/ים הבא/ים. הפריטים והממשקים מפורטים בטבלאות המלוות כל תרשים.

יש לערוך תרשימ/י רכיבים (Component Diagram(s)) בהתאם לצורך. מומלץ לערוך תרשים נפרד לכל פריט תוכנה או צומת חומרה, אולם אם התוכנה איננה גדולה ניתן לערוך תרשים אחד לכולה.

### ארכיטקטורה לוגית ל<פריט/צומת/תוכנת> <שם הפריט/הצומת/התוכנה>

Component Diagram

איור 3: הארכיטקטורה הלוגית (רכיבים וממשקים) של <פריט/צומת/תוכנת> <שם הפריט/הצומת/התוכנה>

טבלה 5: טבלת רכיבים וממשקים ל<פריט/צומת/תוכנת> <שם הפריט/הצומת/התוכנה>

| **רכיבים** | | | **ממשקים** | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **זיהוי** | **שם** | **תפקיד** | **זיהוי** | **סוג** | **שם** | **פירוט** |
|  |  |  |  | <מסופק/נדרש> |  |  |
|  | <מסופק/נדרש> |  |  |
|  | <מסופק/נדרש> |  |  |

## ארכיטקטורה כוללת

הארכיטקטורה הכוללת של המערכת (ר' איור 4) מתארת את צמתי המערכת הפיזית, הכוללים ports של הממשקים הפיזיים, ואת הארכיטקטורה הלוגית של רכיבי זמן-אמת המותקנים בצומת ואת ההאצלה (delegation) בין ממשקיהם הלוגיים ל-ports הפיזיים.

Composite Diagram

איור 4: הארכיטקטורה הכוללת של <פריט/צומת/תוכנת> <שם הפריט/הצומת/התוכנה>

# מפרט תכן תוכנה

## מודל מחלקות

*יש לבנות מודל מחלקות (Class Diagram) של המחלקות העיקריות (סדר גודל: 15-30) מהן בנויה התוכנית. כל מחלקה תכיל פירוט של המאפיינים שלה (attributes, data-members) ושל המתודות שלה (methods, member-functions), תוך ציון מידת הפרטיות של כל אחד ממרכיבים אלה (private, public, protected). מומלץ לחלק את המחלקות למארזים (packages) ולבנות תרשים מחלקות (Class Diagram) נפרד לכל מארז. במידה ויש זיקה כלשהי (association) בין מחלקות ממארזים שונים ניתן לשכפל את המחלקות הרלוונטיות בתרשימים שונים. אין לכלול בתרשים מחלקות שמקורן בספריות קיימות (כגון מחוללי GUI וכו'). במידה ומחלקות חדשות יורשות מחלקות ממקור חיצוני אין לכלול את האלמנטים (מאפיינים ומתודות) שעברו בירושה (אפשר להגדיר לכלי ה-UML לעשות זאת).*

### תרשים מחלקות למארז/רכיב <שם המארז/הרכיב>

Class Diagram

איור 5: תרשים מחלקות של מארז/רכיב <שם המארז/הרכיב>

## מודל התנהגות

*יש לערוך תרשימי רצף (Sequence Diagrams) עבור ה- Use Cases שפורטו בסעיף 4.4. תרשימי הרצף צריכים לכסות את תרשים ההצלחה הראשי ואת כל ההסתעפויות. האובייקטים המשתתפים בתרשים הם מהמחלקות שפורטו בסעיף 6.1 וההודעות ביניהן הן המתודות שהוזכרו שם. מומלץ שהרצפים לא יהיו "חוצי צמתים", כיוון שאז לא ניתן לקרוא ישירות למתודות ויש להעביר הודעות בתקשורת.*

### תרשימי רצף ל-Use Case <שם ה-UC>

Sequence Diagram

איור 6: תרשים רצף של <תרחיש/ים>

# יישום ובניה (מדריך למתכנת)

*יש לפרט את כלי היישום: קומפיילרים, דבגרים, סביבות פיתוח וכו', הנדרשים לצורך יישום התוכנה. יש לצרף תרשים של מבנה התוכנה (קבצים) והנחיות מפורשות כיצד לבצע בניה (build) של התוכנה לצורך התקנה והפעלה. במידה ויש הנחיות או הוראות מיוחדות למתכנתים לגבי כתיבת הקוד והפקת קבצי הביצוע יש לפרטן כאן.* ***זיכרו: תוצר הפרויקט שלכם נשאר בחברה ועשוי לעבור פיתוח המשך, שינויים ותיקונים, לכן יש להשאיר בידי החברה את כל המידע הנדרש לשם כך!***

# מפרטי בדיקות

*מפרטי הבדיקות מספקים הנחיות לביצוע בדיקות שיוכיחו שהמערכת/התוכנה עומדת בדרישות שנקבעו. חובה לכתוב מפרט בדיקות קבלה וניתן לכתוב מפרטי בדיקות גם לבדיקות שילובים חלקיים. מומלץ לכתוב את מפרט הבדיקות על פי הפורמט המקובל בחברה בה מתבצע הפרויקט. ניתן לכתוב את מפרט הבדיקות כמסמך נפרד ולהפנות אליו מכאן או להפיקו כפלט מכלי אחר (למשל כלי לניהול בדיקות). בהיעדר פורמט כזה יש לערוך את מפרט הבדיקות בהתאם לתת-הסעיף הבא:*

## מפרט בדיקות <שם סבב הבדיקות, למשל קבלה>

### התוצר הנבדק

*יש לפרט מהו התוצר הנבדק(רכיב תוכנה, מערכת, תת-מערכת וכו') ולציין את זיהויו כמופיע במפרטים.*

### סביבת הבדיקות

*תיאור הסביבה בה מתבצע סבב הבדיקות. מומלץ להשתמש בתרשים, ובעיקר בתרשימי פריסה (Deployment Diagrams) ותרשימי רכיבים (Component Diagrams) שיראו בבירור את התוצר הנבדק ואת רכיבי הבדיקה/הדיווח/ההפעלה שבסביבתו, כגון drivers, stubs, כלי בדיקה, סימולטורים וכו'. יש לצרף הסבר קצר לגבי כל אחד מרכיבי הבדיקה.*

### פירוט הבדיקות

*יש לפרט את תרחישי הבדיקה, את הדרישות הנבדקות הנחיות ברורות לביצוע הבדיקה ופירוט התוצאות המצופות.*

# התקנה, שימוש ותיקוף

## התקנה

*הוראות מדויקות כיצד יש להתקין את המערכת/התוכנה ולוודא שהיא מותקנת כהלכה. ניתן לכתוב את הוראות ההתקנה במסמך נפרד ולהפנות אליו מכאן.*

## שימוש

*יש לציין מיהם משתמשי התוכנה (למשל, מפעיל, טכנאי וכו') ולצרף הנחיות הפעלה ברורות לכל סוגי המשתמשים. מומלץ לכתוב את הוראות השימוש במסמכים נפרדים (כגון מדריך למשתמש, מדריך למתחזק וכו').*

## תיקוף

*במידה ותוצר הפרויקט אמור לעבור הפעלה נסיונית ותיקוף בשטח (למשל גרסת ביתא) יש לפרט את סביבת התיקוף, משתמשי התיקוף ומשך התיקוף המתוכנן.*

# סימוכין

*רשימה ממוספרת של מסמכים ומקורות אחרים. בכל מקום בגוף המסמך בו קיימת התייחסות לאחד ממסמכים אלה יש לציין את מספרו בין סוגריים מרובעים.*

1. מסמך ייזום לפרויקט <שם הפרויקט> , גרסה <מספר גרסה>.