

# Software Quality Assurance - אדסטה אינה גונג

## מטרת הקורס:

אבטחת איכות תוכנה מהווה חלק בלתי נפרד מעולם פיתוח התוכנה, והיא הסמכות המקצועית המאשרת את תקינות הגרסה ומעורבת בכל שלבי הפיתוח ובפרט טרם השקתו לייצור. בשנים האחרונות חל שינוי גדול בתפקיד ה-QA חברות רבות מכירות בחשיבות ובצורך של אנשי QA מוכשרים וטובים וע"כ שידרגו את מעמדם מבחינת המשכורת ואפשרויות קידום. וכן הורחבו מס' התקנים של אנשי QA בפרויקט פיתוח התוכנה. ומנגד עלו הציפיות מה-QA הן מבחינה מקצועית – הכשרת ISTQB וידע טכנולוגי רחב, והן מבחינת כישורים ואינטליגנציה.

ISTQB – גוף בינלאומי להכשרת בדיקות תוכנה.

## תכני הלימוד בקורס:

✓ SQA – מדוע ולמה?

✓ שילוב אבטחת איכות בתהליך הפיתוח.

✓ <sup>קביעת סט</sup> <sup>דגימות</sup> <sup>דינאמי</sup> Dynamic testing & Static testing

✓ <sup>קביעת סט</sup> <sup>פונקציונלי</sup> <sup>קביעת סט</sup> Functional Testing & Nonfunctional Testing

✓ <sup>דגימת קוד</sup> <sup>חינוך/תמיכה</sup> Unit Testing

✓ Test first Approach - פיתוח מובל בדיקות

## מבוא לבדיקות תוכנה

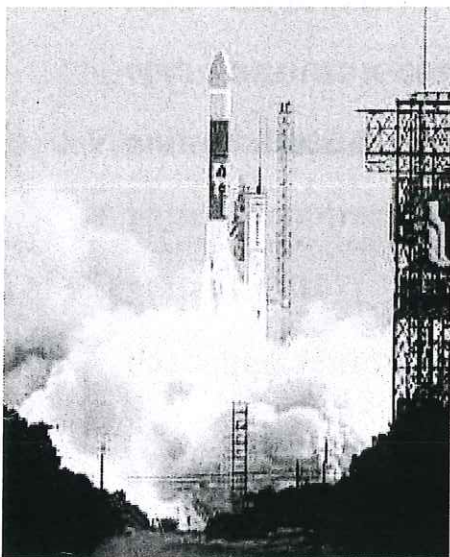
### באגים ונזקים

בעשרים השנים האחרונות טכנולוגיות המחשוב התפתחו והשתלטו כמעט על כל היבט בחיינו: עסקאות בנק, טלפונים, מכוניות וכו'. בהתאמה – מערכות התוכנה נעשו מורכבות יותר ולכן גם פחות יציבות, והתוצאה היא: תוכנות לא טובות שגורמות נזקים, גם כספיים ואפילו נזקים בנפש. הנה לפניך מספר דוגמאות רק כדי להמחיש:

מערכת הניווט של מעבורת חלל תוכנה לפי יחידות של אינצ'ים במקום יחידות של ס"מ, דבר שגרם לנזק עצום.

מערכת המכס בשדה התעופה של לוס-אנג'לס לא עבדה נכון, וגרמה להשבתת שדה התעופה למספר רב של שעות. 17,000 טיסות עוכבו.

מערכת המנהלת את מכשיר ה CT -בבית-חולים לא עבדה נכון וגרמה למותם של 5 אנשים. והרשימה עוד ארוכה...



כך למשל קרה ב-1999 באוקטובר כאשר החללית האמריקאית **Orbiter Climate Mars** אבדה בחלל. החקירה גילתה שטעות תוכנה קטנה, שגרמה לשימוש במידות מסוימות לפי יחידות מידה אנגליות במקום יחידות מידה מטטריות גרמה לאובדן החללית ולנזק של 125 מיליון דולר.

בדומה ב-1996 טיל אריאן בשווייץ מיליארד דולר של סוכנות החלל האירופאית התרסק דקה לאחר שיגורו עקב באג בתוכנת הניווט של הטייל.

ואילו באגים במכונת ההקרנה הרפואית Therac-25 גרמו למות המטופלים עקב רמת קרינה שגויה.

בארצות הברית לבד מוערכים נזקי הבאגים השנתיים בקרוב ל \$100,000,000,000

## What is Quality??

## הגדרה:

[illegible]

## מאפייני האיכות:

3. ציגה געשקט'ן ערן פאר געזונט, ביזנאס פאר טיג, געזונט און וואלל, מוסר פון געזונט.  
 4. געזונט און געזונט פאר מוסר, געזונט און געזונט, געזונט און געזונט, געזונט און געזונט.  
 5. געזונט און געזונט פאר מוסר, געזונט און געזונט, געזונט און געזונט, געזונט און געזונט.

## מטרות לאיכות:

[illegible]

1.178 100 100 100 100 100

האיכות משתלבת בכל שלב במחזור החיים של

תהליך הייצור (מוצר או שירות):

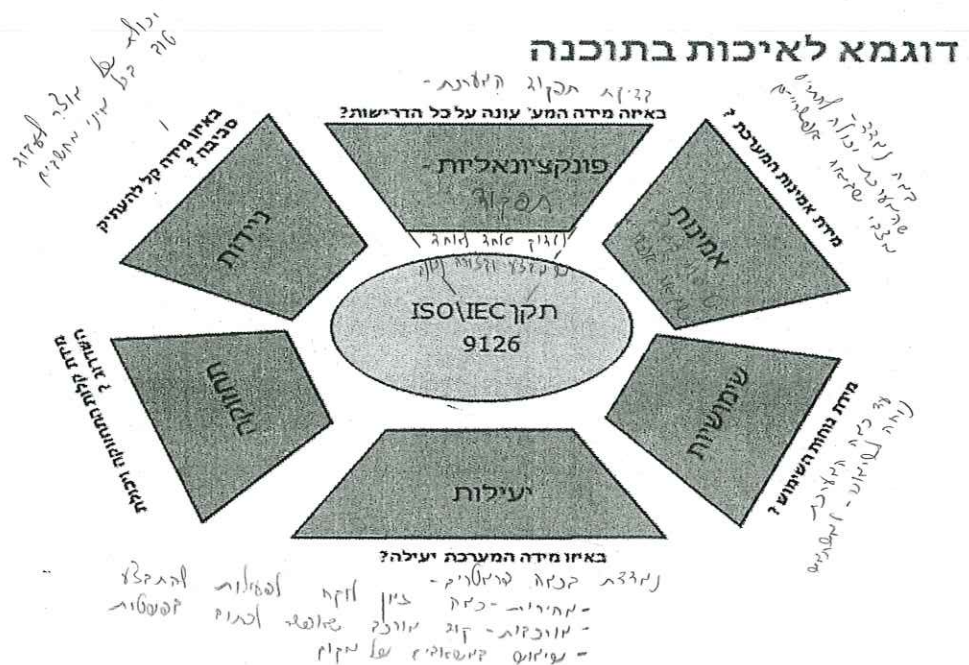
ניתוח הבעיה - /על מה המציגים את המוצר האמיתי תחת השם האמיתי. /

עיצוב - הסברה איך יושא המזלז

מימוש - ת"צ, גרונתה - של ת"צ, קה"צ

בדיקה - תוצאות המבחן

תחזוקה - עיבודי



## בדיקות ואיכות:

- בעזרת הבדיקות ניתן למדוד את איכות התוכנה (ע"פ מס' התקלות) *נאמנה לא אה ג'יין*
  - בתהליך הבדיקות בוחנים את התוכנה מבחינה פונקציונאלית ולא פונקציונאלית. *הפונקציונליות*
  - ROI (Return On Investment) ע"י הבדיקות מפחיתים את הסיכונים של השימוש בתוכנה *הסיכונים*  
"ולשמור את הזמן ואת הכסף". - אל הגטע - שיחצי את מה שהקצי
  - הבדיקות מגבירות את האמון בתוכנה. - גינה לעיני יור אציון
- ניתוח התקלות והפקת הלקחים משפרים את המערכת הבאה. - אכסר אהרן מיקו גזון, אהרן אהרן, אהרן אהרן

## אבטחת איכות = בקרה על התהליכים

פעולות שמטרתן להבטיח איכות בכל אחד משלבי מחזור החיים של פיתוח התוכנה, ומתמקדת בתהליך המוצר השלם.

שמטרתן להבטיח את איכות המוצר ואיכות התהליך  
ע"י נהלים, תקנים, הדרכות בדיקות ומדדים ..

## QC בקרת איכות = בקרה על המוצר

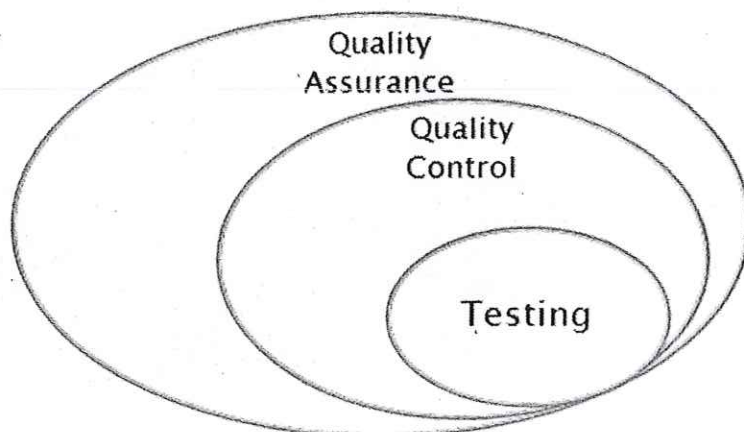
יישום של הנהלים ותקנים שהוגדרו ע"י צוות אבטחת האיכות.

בדיקת המוצר אל מול הדרישות שהוגדרו.

ע"מ לוודא שהמוצר עונה על הדרישות. - Validation

וזה כולל: בדיקות סטאטיות ובדיקות דינאמיות לסוגם בדיקות פונקציונאליות ולא-פונקציונאליות.

Quality Assurance, Quality Control & Testing





## מדוע נגרמות שגיאות?

[illegible]

מה מקור הבאג??

**Error - שגיאה** פגום או שגיאות מוצאות לא רצויות, שווים וקבוע - גדולות.

Defect - פגם

00'00 שניות    בנייה של תוכנית    התכנון והעבודה    הקמת התשתית    הקמת המבנה    הקמת הציוד    הקמת ההפעלה

**דוגמא:** words משמעות מילוק לטובת - מילוקים הם זה (קצת) כמו מילוקים

## Debugging & Testing

22. האם יש צורך בשינוי מדיניות הממשלה בנושא? **Testing**

י'ס' - י'ס' - Debugging

ההבדלים: Confirmation testing  
במקור על מנת

## ההבדלים:

[illegible]

דביקת ינארה אהבה מן אל מעונת קוץ חסור debug א חיוב

קציקור שם יד מ31/12/18 גסיסלע, און און debug פאלט זיין :

סכמי בהתייחס לפרויקט תוכנה באופן כללי:

a. פרטי מה בודקים?

מגן שמש בקומה הראשונה, רחוב, מדרום שלל, הרצליה 8  
סניף עיריית תל אביב-יפו, מדרום שלל, הרצליה 8

b. כמה פעמים בודקים?

מה שיש

c. מתי מסתיימות הבדיקות? - (שאלה) עלויה מצליח ותקין עדיין, נמוכה מעליו, מצליח, רכש קטל, גיליון המענה

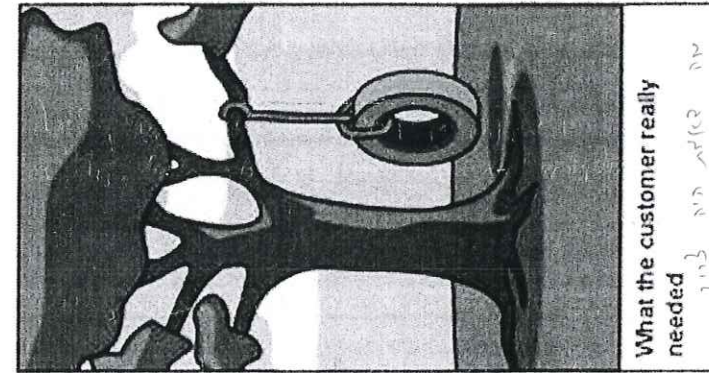
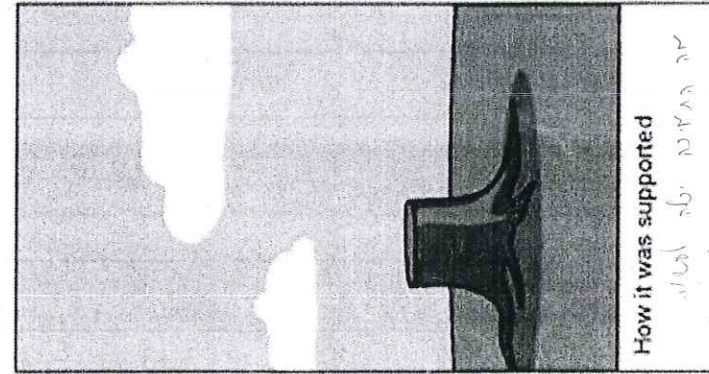
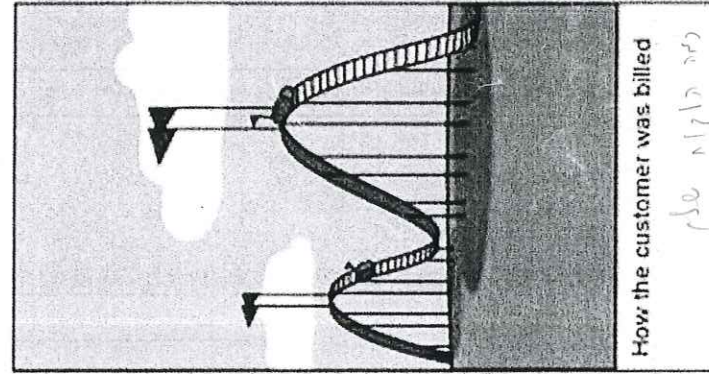
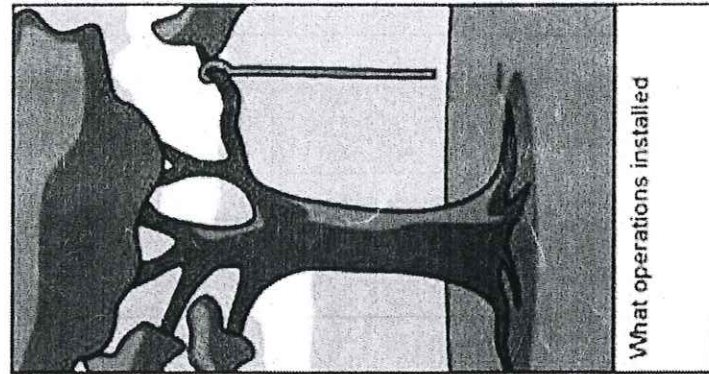
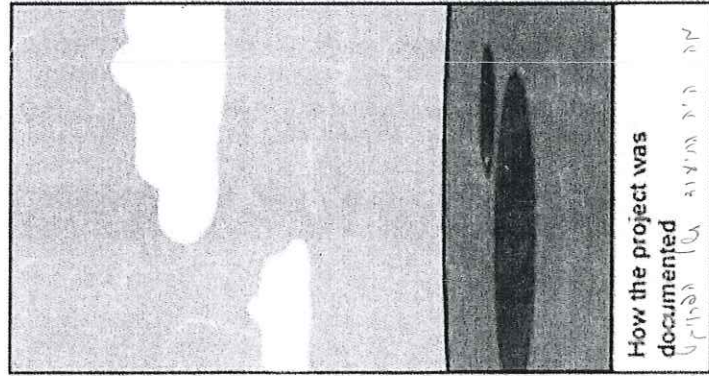
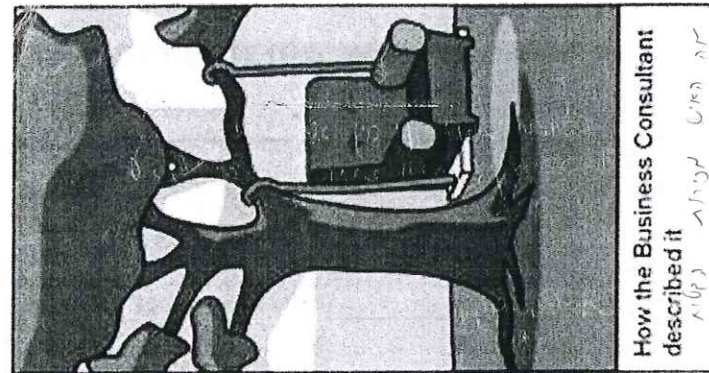
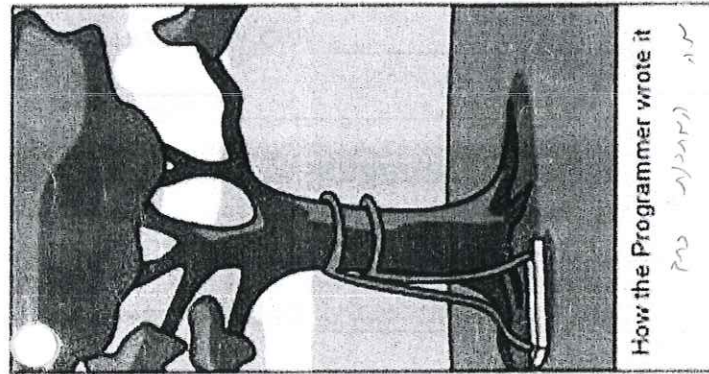
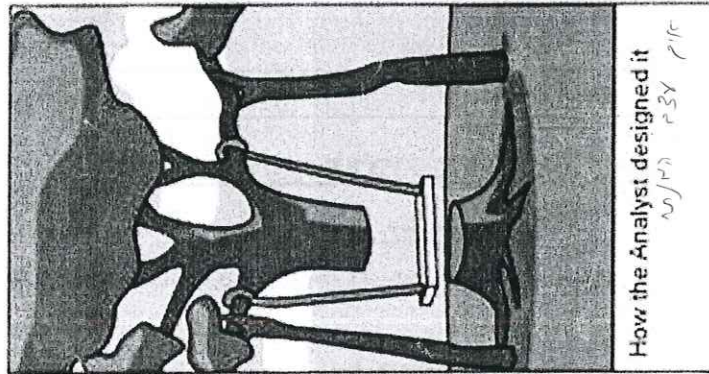
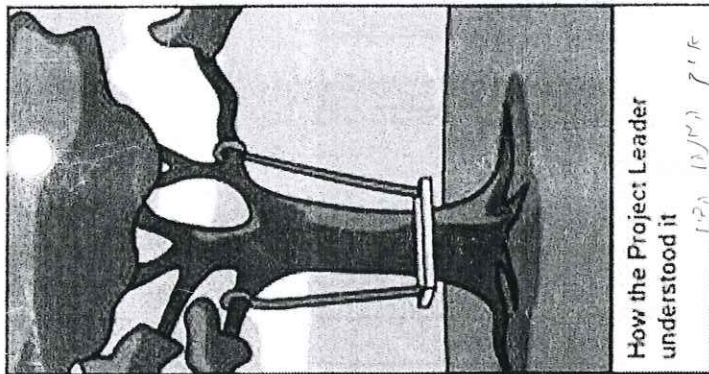
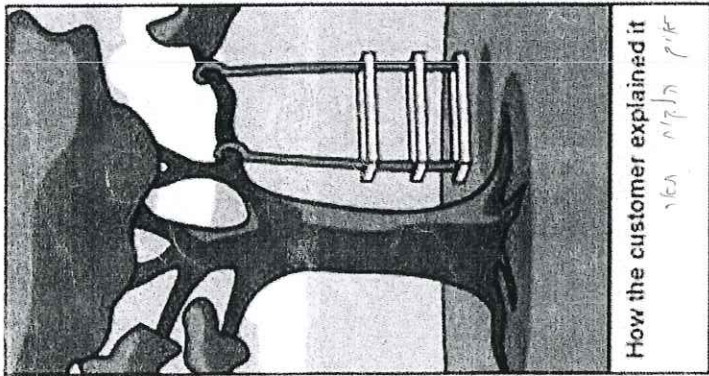
[illegible]

P.3, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 (2)

d. הגורמים לתקלות בתוכנה?

about we are, right?







# קורס SQA

## תהליכי בדיקות ומסמכים

תכנון וקריה STP – Planning and Control- ↓

נהיגה וז.צ.ק STD – Software Design – ↓

הוצאה וקריה STR – Test Execution – ↓

## תהליך הבדיקות

החלק המשמעותי והעיקרי בתהליך הבדיקות הוא שלב ביצועם. אולם, על מנת לבצע בדיקות יעילות, יש להשקיע זמן בתכנון הבדיקה, הגדרתה והכנת שדה הבדיקות לביצועה.



- תכנון ובקרה. מסמך הקריה
- ניתוח ועיצוב. קישור הקריה
- יישום וביצוע. האמצעים להדגמה
- ניתוח תנאי יציאה וביצוע דיווח. אסטרטגיה הקריה
- פעולות בגמר הבדיקות. אחראי צוות

הסבירי software test description - גאון קריה גאון

STD .a Software test Plan קריה גאון

TRD .b test Requirement Definition קריה גאון

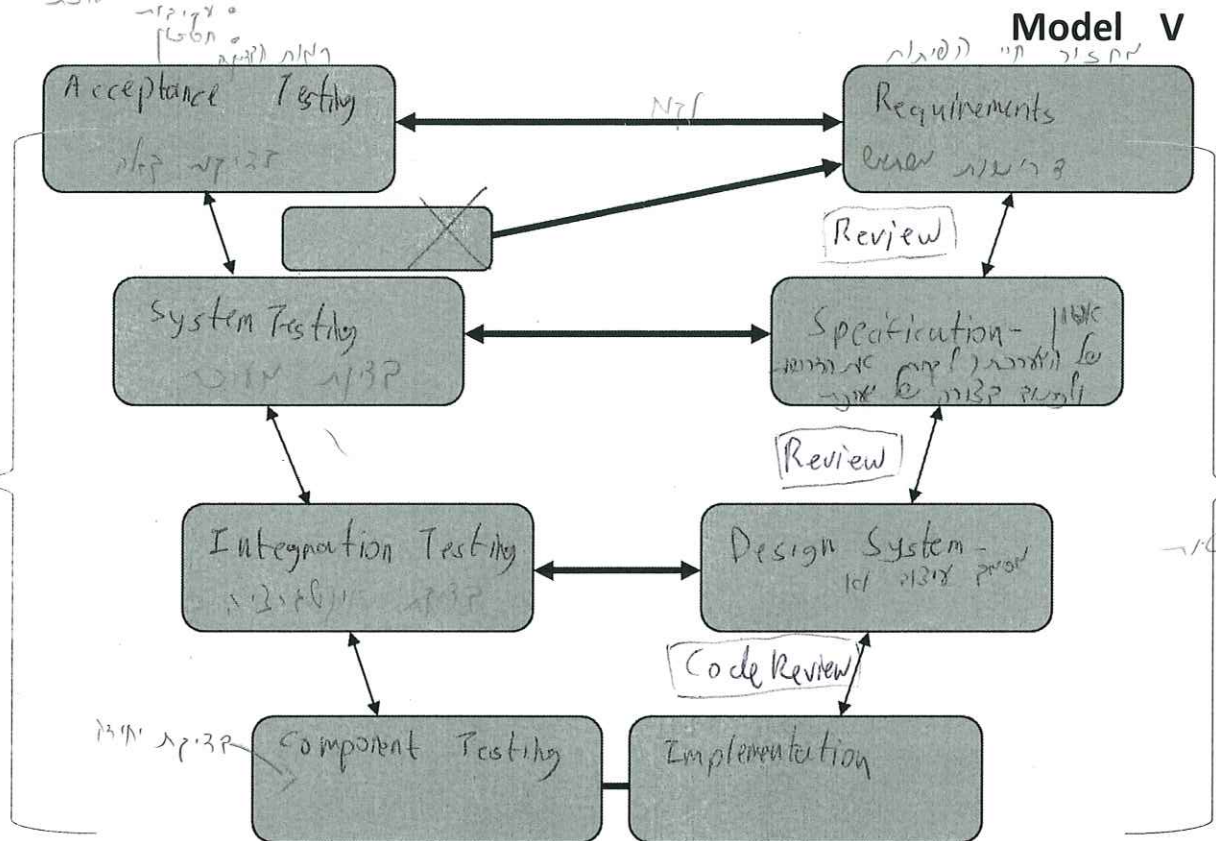
STR .a Software test Report/Result - קריה גאון

## Verification & Validation V&V

Verification	Validation	הסבירי/תרגמי
איוש	תקוף	סוג בדיקות נדרשות
קריה סטטית - Review	קריה פונקציונלית - טיפוסית	מה בודקים?
איוש	איוש	בפני מי?

# קורס SQA

יחידת המבחן:  
 • דגש על תוכנית המבחן  
 • תוכן ודיווח הדיווח  
 • הדיווח המבצע קריאה לפרט  
 • ודיווח המבחן  
 • ודיווח המבחן  
 • ודיווח המבחן



## סיכום בדיקות לפי ISTQB

**Dynamic testing** בדיקות קוד אנו מנסים את המערכת עצמה או את חלקים

**Static testing** בדיקות המבצעים בדיקות אנו מנסים את הקוד או מנסים את המבחן

ההבדלים:

1. מבנה של מידע וקריאה בדיקות על מבנה המערכת (בדיקות מוקדמות-בדיקות בדיקות) וקריאה  
 2. מבנה של מידע וקריאה בדיקות על מבנה המערכת (בדיקות מוקדמות-בדיקות בדיקות) וקריאה  
 3. מבנה של מידע וקריאה בדיקות על מבנה המערכת (בדיקות מוקדמות-בדיקות בדיקות) וקריאה

המטרה:

מבנה של מידע וקריאה בדיקות על מבנה המערכת (בדיקות מוקדמות-בדיקות בדיקות) וקריאה

מבחן אפיון מבנה של מידע וקריאה בדיקות על מבנה המערכת (בדיקות מוקדמות-בדיקות בדיקות) וקריאה

המטרה של מבנה של מידע וקריאה בדיקות על מבנה המערכת (בדיקות מוקדמות-בדיקות בדיקות) וקריאה

בדיקות - מבנה של מידע וקריאה בדיקות על מבנה המערכת (בדיקות מוקדמות-בדיקות בדיקות) וקריאה

מבחן אפיון מבנה של מידע וקריאה בדיקות על מבנה המערכת (בדיקות מוקדמות-בדיקות בדיקות) וקריאה



# קורס SQA

## רמות בדיקה

- ✓ בדיקות יחידה / רכיבים ( Unit Tests \ Component Testing )
- ✓ בדיקות אינטגרציה Integration Tests
- ✓ בדיקות מערכת System Tests
- ✓ בדיקות קבלה Acceptance Tests

## בדיקות יחידה \ Component \ Unit Tests

**מטרה :** למצוא פגמים ברכיבים הניתנים לבדיקה באופן נפרד/ עצמאי.

**מבוצעות ע"י :** המפתח בסביבת הפיתוח, עבור רכיב שפותח על ידו.

**מה נבדק:** בדיקת פונקציונאליות ולא פונקציונאליות של רכיב בודד (מסך/ דו"ח).

**שיטת הבדיקות :** בדיקה תוך שימוש בסימולטורים כדי לדמות קלט (INPUT) ע"י מנהל התקנים (Driver) או פלט (Output) ע"י קבוע (STUB)

## רמות בדיקה – אינטגרציה Integration Tests

**מטרה :** בדיקות ברמת הרכיב או ברמת המערכת שמטרתן בדיקת ממשקים בין רכיבים ובין ונוערכות וכן יחסי גומלין בין חלקיה השונים של מערכת.

לדוגמא: הפעלה, קבצים, חומרה, תהליכים, E2E

**מבוצעות ע"י :** המפתח ואו בודק תלוי בשלב האינטגרציה בסביבת האינטגרציה

**מה נבדק:** בדיקות שילוב של רכיבים, תתי מערכות ומערכות

**שיטת הבדיקות :** Big bang, Top down, Button up

## רמות בדיקה – מערכת System Testing

**מטרה :** בדיקות המתייחסות למערכת כמכלול, מתבצעות על סביבה התואמת ככל האפשר לייצור ומבוצעות ע"י צוות בדיקות עצמאי, במטרה לבדוק תאימות המערכת לדרישות.

**מבוצעות ע"י :** הבוודקים בסביבת הבדיקות ( סביבה דומה ביותר לסביבת הלקוח)

**מה נבדק:** בדיקת המערכת כמכלול לפני המסירה ללקוח, בדיקה שהמערכת עונה על הדרישות ועובדת נכון ללא תקלות.

**סוג הבדיקות :** בדיקות פונקציונאליות כמו תהליכים E2E, אבטחת מידע ובדיקות לא פונקציונאליות כגון: עומסים, ביצועים, כשל והתאוששות ועוד.

**שיטת הבדיקות :** ערכי קצה, קבוצות שקילות, טבלת החלטה מקרי שימוש ומעבר בין מצבים.

# קורס SQA

## רמות בדיקה – קבלה Acceptance Testing

**מטרה:** בדיקות שמטרתן לבסס ביטחון במערכת או בחלקים ממנה. בדיקות קבלה יכולות להתבצע בזמנים שונים במחזור החיים של הפרויקט ( לפני התקנה או שילוב במערכת קיימת, בדיקת רכיב, או שיפור תפקודי לפני בדיקות מערכת). מבוצע ע"י הלקוח או מי מטעמו, על סביבה תואמת ייצור על מנת לאשר שקיבל את המערכת לשביעות רצונו.

**מבוצעות ע"י:** הלקוח והמשתמשים בסביבה המבצעית (האמתית) בזמן קבלת המערכת

**מה נבדק:** בדיקת המערכת אל מול הדרישות.

**סוג הבדיקות:** סוגי הבדיקות השונים הם בהתייחס למשתמשים השונים או המצבים שבהם צריכה המערכת לעמוד

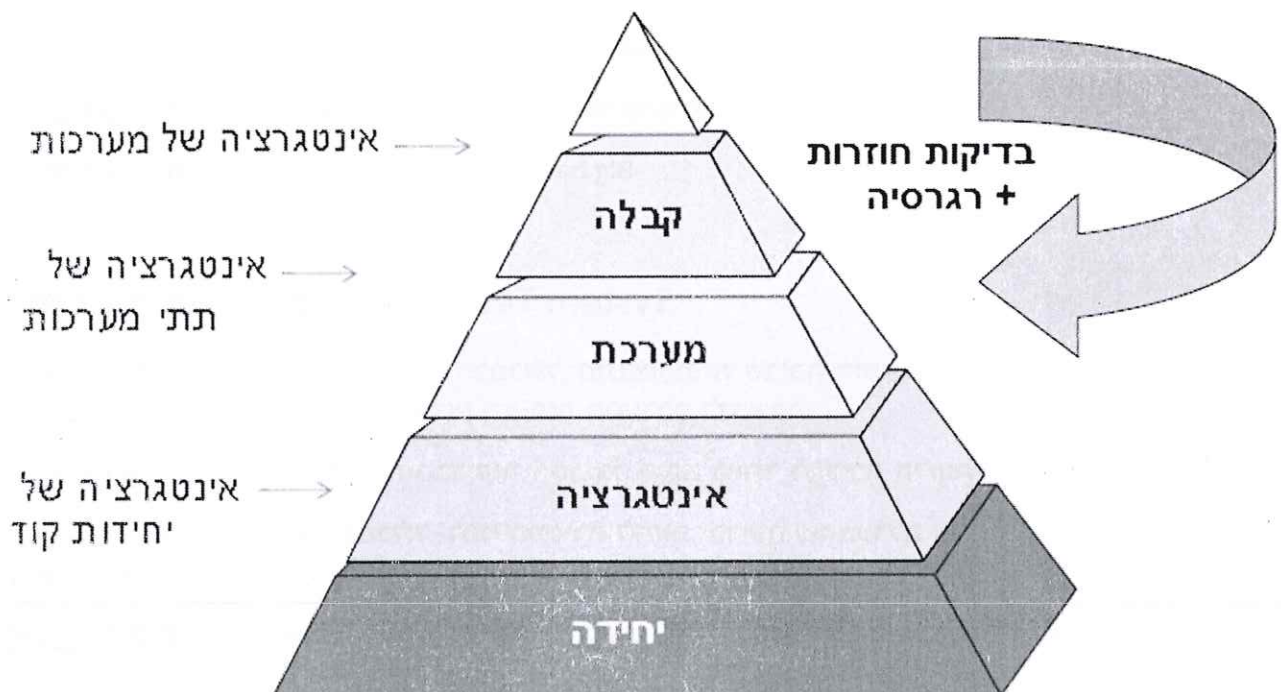
**בדיקות משתמש –** מאמתות את כשירות המערכת לשימוש בידי משתמש עיסקי

**בדיקות תפעוליות –** קבלת המערכת ע"י מנהלי המערכת כולל: גיבוי ושיחזור, התאוששות מאסון, ניהול משתמשים, משימות תחזוקה, עומס נתונים, בדיקות תקופתיות לפריצות אבטחה.

**בדיקות חוזיות ורגולטוריות –** בדיקות עמידת המערכת בחוקים ותקנים (ממשלתי/ משפטי/ תקני בטיחות)

**בדיקות שטח אלפא –** באתר הפיתוח על ידי צוות הבדיקות

**בדיקות שטח ביתא –** בדיקות שדה על ידי לקוחות פוטנציאליים במקומות בהם הלקוחות נמצאים.





# קורס SQA

## סוגי בדיקות :

Black Box - קופסא שחורה - לא רואים את הקוד הפנימי

- בדיקות פונקציונאליות

- בדיקות לא פונקציונאליות

- תפקוד התוכנה קלט לעומת פלט (ללא התייחסות לקוד)
- הפעלת המערכת כפי שתופעל ע"י המשתמשים על בסיס פעולה ותגובה
- בדיקות ביצועים
- בדיקות כשל והתאוששות
- בדיקות MTBF = משך הזמן שעובר בין תקלה אחת לבאה. = מדד לאיכות.

White Box - בדיקות תלויות מבנה גלוי

- מיקוד בקוד ובמהלך עיבוד הנתונים : בדיקת מסלולים , תנאים , זרימת מידע ולולאות בתוך הקוד.
- ניתוח חוקיות בשפת הפיתוח, כיסוי לוגי Flow Graph דרך כל קטעי הקוד, בדיקת איכות הכתיבה ע"י Code Review.

### בדיקות שינויים

- בדיקות אישור ורגרסיה (ראו לעיל)

- בדיקות תחזוקה

בדיקות שמבוצעות על מערכת שכבר פועלת אצל הלקוח.

✓ לאחר עליה הראשונית לאויר , המערכת תהייה בשימוש שנים רבות ועוברת שינויים ועדכונים רבים.

# קורס SQA

10/08/2018 - 10/08/2018

## 1.1 פירוט סוגי בדיקות התוכנה הקיימות:

- **בדיקות פונקציונליות (Functional)** - לאימות פעילות המערכת. בדיקות אלו מבוססות על מסמך הדרישות ומסמך האפיון ומטרתן לבדוק כי המערכת עושה את מה שהיא צריכה ולא עושה את מה שאינה צריכה לעשות (valid and invalid testing)

### דוגמא לבדיקה

- **בדיקות ממשק לקוח (GUI)** - בדיקות הפקדים והשדות במסך. התנהגות תקינה, פורמט של שדות, בהתאם לחוקיות המוגדרת ברמת המסך ולא הלוגיקה העסקית. לדוגמה: בדיקת מינימום ומקסימום תווים בשדה. *לדוגמה: בדיקת מינימום ומקסימום תווים בשדה. *לדוגמה: בדיקת מינימום ומקסימום תווים בשדה.**

### דוגמא לבדיקה

- **בדיקות תפעול (Activation)** - בדיקת התנהגות המערכת באופן לוגי ומבוקר. לדוגמא: מעבר מ-Mode מבצעי ל-Mode תחזוקה או ל-Mode הדרכה.

### דוגמא לבדיקה

- **בדיקות ממשקים (Interfaces)** - בדיקת ממשקים פנימיים וממשקים חיצוניים מהמערכת ואליה לדוגמא: שירותי בנקאות אוטומטיים, מערכת שליטה ובקרה וכד'.

### דוגמא לבדיקה

- **בדיקת נתונים והסבות (Conversion)** - בדיקת הנתונים (Data) במערכת: בדיקת טיוב נתונים ובדיקת שלמות נתונים - Data Integrity - כמו כן בדיקת הסבת נתונים: הסבת הנתונים מהמערכת הנוכחית לחדשה.

### דוגמא לבדיקה

- **בדיקות שימושיות (Usability)** - בדיקות נוחות השימוש ויעילות העיצוב של האפליקציה ונגישות לבעלי מוגבלויות. לדוגמה: נוחות השימוש בתפריטים, ניווט נוח והתמצאות באתר.

### דוגמא לבדיקה

- **ביצועים ועומסים (Performance)** - בדיקת המערכת מתוך מטרה לוודא כי המערכת מתפקדת כנדרש במצבי עומס שכיחים וקיצוניים כמו כן בדיקת תפקוד המערכת בעומס גבוה לאורך זמן או בנקודות זמן קצרה.

### דוגמא לבדיקה



# קורס SQA

**בדיקות אבטחת מידע (Security)** - בדיקת המערכת מתוך מטרה לאתר כשלים במערך ההגנה על המידע במערכת: בדיקת מערך ההרשאות והמידור, בדיקת סיסמאות - מבנה ותוקף ובדיקות חדירה. **דוגמא לבדיקה**

---

---

**בדיקות שרידות והתאוששות (Survival and Recovery)** - בדיקת מנגנון ההתאוששות מכשל חומרה ותוכנה, בדיקת רציפות פעולת המערכת לפרק זמן ממושך ומידת התאוששותה מכשל, הרצת מדגם תהליכים עסקיים ברציפות לפרק זמן ממושך ובדיקת מערך הגיבויים והשחזורים במערכת.

**דוגמא לבדיקה**

---

---

**בדיקות התקנה (Installation)** - בדיקת תהליך התקנת המערכת בסביבת הייצור ובדיקת התקנת מערכות צד שלישי.

**דוגמא לבדיקה**

---

---

**בדיקות תאימות (Compatibility)** - בדיקות לאימות יכולת שילוב התוכנה/רכיב תוכנה במערכת קיימת/חדשה (למשל - תאימות של התוכנה לעדכוני מערכת הפעלה, דפדפנים שונים, תוכנות אחרות שהתוכנה אמורה לעבוד בשילוב עימן, וכדומה).

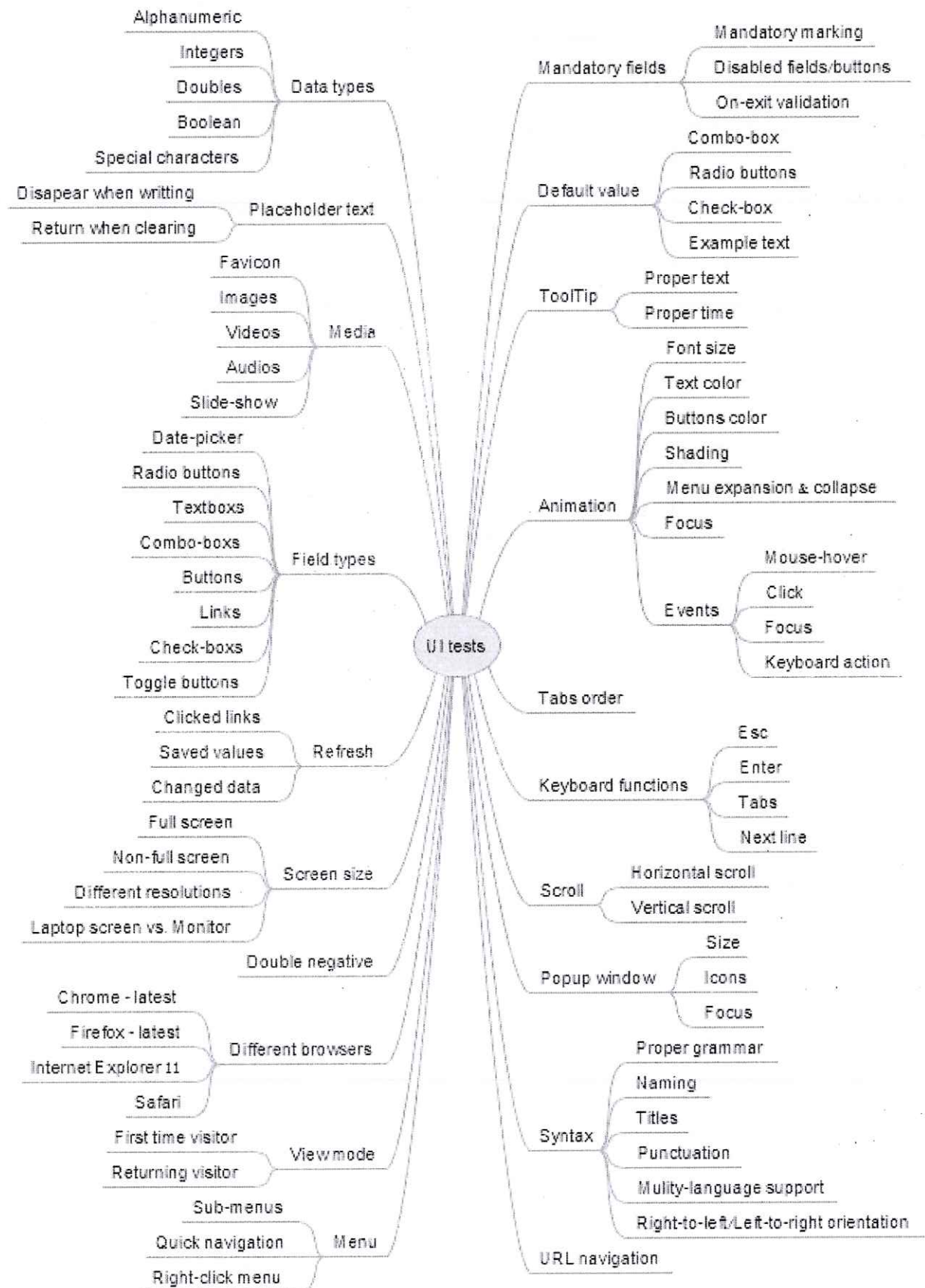
**דוגמא לבדיקה**

---

---

# קורס SQA

## בדיקות-GUI





## תרגיל סוגי בדיקות

**אבטחת מידע** – המטרה: שמירה על סודיות ואמינות המידע שלך.

מתן השירותים הבנקאיים באמצעות קווי תקשורת, לרבות ערוצי תקשורת שאינם מוצפנים, כרוך, מעצם טיבו, בסיכונים שונים, לרבות חדירה למערכות המחשב לשם גניבת מידע, שינוי או מחיקתו, השתלטות על המערכות של הבנק, הקמת אתרים מתחזים, הונאה מקוונת, התחזות לבעלי החשבון, גניבת פרטי זיהוי, פעילות בשם בעלי החשבון, חשיפת מידע ועוד.

הבנק נוקט ו/או ינקוט באמצעים שונים, לפי שיקול דעתו, לצמצום הסיכונים האמורים. בנוסף, הבנק ממליץ לבעלי החשבון להכיר את הסיכונים הגלומים

בשירותים הבנקאיים באמצעות קווי תקשורת ואת האמצעים הנדרשים על מנת להגן מפני סיכונים אלו, ולפעול לצמצומם, ובין היתר לפעול על פי הנחיות הבנק.

**לשם זיהוי המשתמש, ייעשה שימוש באמצעי זיהוי.**

**באמצעות אמצעי הזיהוי יוכל כל משתמש לקבל את כל המידע ביחס לחשבון, וכן לבצע כל פעולה בחשבון, לפיכך קיימת חשיבות עליונה לשמור על אמצעי הזיהוי.**

## תרגול יצירת סיסמא:

על מנת ליצור סיסמה מורכבת, בטוחה ושאינה קלה לפיצוח, מומלץ להקפיד על מספר כללים פשוטים:



- על הסיסמה להכיל בין 6-8 תווים.
- על הסיסמה להכיל שילוב של אותיות ומספרים.
- אין להשתמש ברצף של תווים חוזרים כגון: 111.א.א.א
- אין להשתמש בתווים עוקבים כגון: 123 או ABC.
- אין להשתמש בתווים סמוכים במקלדת כגון: QAZ:א.א.SDF
- אין להשתמש בסיסמאות שבהן השתמשתם בעבר.
- הימנעו מסיסמאות קלות לניחוש כגון מס' תעודת הזהות, תאריכי לידה, שמות של בני זוג או ילדים וכד'.
- אל תשתמשו בקוד המזהה (שם משתמש), או בחלק ממנו, כסיסמה.