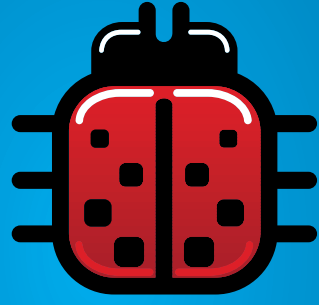


דבעון רביעי 2021 גיליון מס' 27

מגזיר

# עולם הבדיקות



[www.testingworld.co.il](http://www.testingworld.co.il)

ניתוח אפיון

ליאור שם טוב

בודקות  
חרדיות - חלק  
ראשון  
מיכל קדושים

AI

ראיון עם  
מנהלת  
בדיקות  
יפעת שריפי

"פריצת" הדרך  
שלי לעולם  
הסקיוריטי  
אסף סהר

בחן את  
עצמך  
טל פאר

# דבר העורך ניצן גולדנברג



## ניצן גולדנברג

מזה 6 שנים בתחום בדיקות התוכנה, תפקיד נוכחי מהנדס בדיקות בכיר בחברת [SeatGeek](#), המוביל הראשי של קבוצת המיטאפ [TestIL](#), מרצה בקורסים לבודקי תוכנה וחבר בצוות המייעץ AB של [ITCB®](#).



## קוראים יקרים,

*"Being told you're appreciated is one of the simplest and most uplifting things you can hear"*

Sue Fitzmaurice

הפעם החלטתי להביא לכם משפט שהוא לא רק קשור לעולם הישיר שלנו "עולם הבדיקות" אלא לכל שלב ושלב בחיים שלנו.

"הערכה"

אין אחד שלא אוהב לקבל הערכה טובה ורובנו אפילו מחפשים לשמוע הערכה טובה מהמנהלים שלנו. הערכה טובה גורמת לנו להרגיש רצויים ונותנת לנו הרגשה שאנו תורמים למקום העבודה ולחברים לצוות. מצד שני, אנו מפחדים לקבל הערכה כי יש מצב שהערכה לא תהיה טובה והיא תשקף ותחשוף צדדים שאנו כאנשים לא אוהבים לשמוע, לדוגמא "אתה עובד מעולה ומקצועי אבל יש לך נטיות לקחת דברים יותר מדי באופן אישי" או "אתה מתפזר על יותר מדי דברים בו זמנית". לעומת זאת יש הערכות שכן כף לקבל כגון "אני רואה שיפור רציני בתפוקת העבודה שלך".

גם למנהלים, תהליך הערכת העובדים אינו תהליך קל. מצד אחד, חשוב לתת הערכה שוויונית וכנה שכן כל דבר שנאמר בשיחת הערכה יכול לבנות ולהפיל את העובד בין אם זו הערכה טובה או הערכה פחות טובה. בשנה האחרונה יצא לי לעשות שיחת הערכה לבודקים ג'וניורים שעובדים איתי וזו לא הייתה משימה קלה. אני ידעתי שעלי לתת הערכות טובות ופחות טובות ולכל עובד כך זה היה. בשיחות הערכה היו לי הרבה דברים טובים לומר על כל אחד ואחת מהם, דבר שהיה לי מאוד קל לומר אבל גם היו לי משובים פחות טובים שהייתי צריך לתת.

קבלת ביקורת זה דבר, שלנו כבני אדם מאוד קשה לקבל אבל חשוב שנדע לקבל את הביקורות הללו בכדי שנוכל ללמוד מהם ולבסוף לפרוח ולהתקדם מהם. בראיון עבודה מבקשים מכם לתת תכונה שלילית אחת וזו שאלה מאוד קשה כי אנו לא רוצים לחשוף את הדברים השליליים שלנו. כאשר נשאלתי באחד מראיונות העבודה שלי איזו תכונה שלילית אני יכול לומר על עצמי, ישר נזכרתי בשיחת הערכה שעברתי עם אחד המנהלים שלי ובה הוא אמר לי שאני לא יודע לקבל ביקורת. באותו הזמן קיבלתי את זה די קשה כי לא אהבתי לשמוע את הביקורת הזו אבל לקחתי את זה למקום שבו אני יכול לפרוח מזה ולהתפתח וכאשר שאלו אותי על התכונה השלילית באותו הראיון עניתי כי התכונה השלילית שלי היא "ביקורתיות" אבל המשכתי ואמרתי שאכן יש לי בעיה של לקבל ביקורת אבל אני עובד על עצמי ולקחתי סמינר של עבודה אישית וכיום אני יודע שהביקורות שאני מקבל על עצמי הן לא למטרת פגיעה בי אלא למטרה של התפתחות עצמית, כי אם יש משהו שמכשיל אותי מלהתקדם אז אני אעשה כל שביכולתי להסיר את המכשול הזה כך שאוכל להתקדם.

לכם קוראים וקוראות יקרים אני אסיים במשפט אחרון, אם אתם מרגישים שלא נותנים לכם מספיק הערכה אז אל תפחדו לבקש אותה, אם אתם מקבלים הערכה פחות טובה, אל תיקחו את זה לפן האישי אלא תלמדו מזה ותראו איך אתם יכולים להשתפר ולהתפתח מזה.

מונח לפניכם גיליון מספר 27 של מגזין "עולם הבדיקות".

בגיליון זה תוכלו למצוא מגוון רחב של מאמרים חדשים, בנוסף לטורים המעולים והקבועים שלנו:

"פריצת הדרך שלי לעולם הסקיריטי" – כתבת מוטיבציה מאת אסף סהר

ליאור שם טוב מביא לנו עוד מאמר מעניין והפעם על "ניתוח מסמך אפיון"

מיכל עדיקה-קדושים באה להציג לנו את העולם של הבודקות החרדיות בסט מאמרים מרהיב אשר ילווה אותנו בגיליונות הקרובים. כמו כן תוכלו להנות מהטורים הקבועים שלנו: ראיון עם מנהלת בדיקות, האנציקלופדיה לבדיקות, מחפש צרות, בחן את עצמך.

אנו נשמח לקבל בקשות לנושאים חדשים ומעניינים למאמרים, צרו עימנו קשר במייל: [Info.testingworld@gmail.com](mailto:Info.testingworld@gmail.com)

קריאה מהנה,  
ניצן גולדנברג



### תוכן העניינים

- 2..... דבר העורך
- 4..... ראיון עם מנהלת בדיקות יפעת שריפי, **Fyber**
- 6..... בחן את עצמך - שאלה | **טל פאר**
- 7..... "פריצת" הדרך שלי לעולם הסקורירטי | **אסף סהר**
- 9..... מחפש צרות | **מיכאל שטאל**
- 12..... בודקות חרדיות - חלק ראשון | **מיכל קדושים**
- 13..... האנציקלופדיה לבדיקות | **קובי יונסי**
- 14..... בחן את עצמך - תשובה | **טל פאר**
- 15..... ניתוח אפיון | **ליאור שם טוב**
- 17..... דף העורכים

מו"ל  
Israeli Testing Certification Board  
ITCB®

ניהול המגזין  
iMDsoft, ברון, יאן

ניהול התוכן  
קובי הלפרין, Nokia

עורך ראשי  
ניצן גולדנברג, SeatGeek

עיצוב גרפי  
בית נלי מדיה  
סטניסלב קולנקו  
[www.beitnelly.com](http://www.beitnelly.com)

יצירת קשר  
אימייל:  
[info.testingworld@gmail.com](mailto:info.testingworld@gmail.com)

הרשמה  
<http://bit.ly/TW-Reg>  
פקס: 03-6176605  
כתובת: ברון הירש 14 בני ברק 51202



[www.testingworld.co.il](http://www.testingworld.co.il)



[www.itcb.org.il](http://www.itcb.org.il)



**עולם הבדיקות נכתב ע"י בודקים  
עבור בודקים**  
**ITCB®** מקדמים את קהילת הבודקים  
בישראל

#### מגזין עולם הבדיקות

עולם הבדיקות הינו מגזין רבעוני. כל הזכויות שמורות. זכויות היוצרים על חומר שפורסם על ידי המפרסם הינן רכושן של המחבר. הדעות המובאות במאמרים והתוכן לא בהכרח משקפים את דעת המפרסם. המחברים הינם האחראים הבלעדיים על תוכן מאמרם. מובהר כי העתקה ו/או נטילה שיטתית של מידע מהמגזין לצורך פעילות מסחרית ו/או עסקית, או לצורך כל פעילות אחרת שיש בה כדי לפגוע בפעילות העמותה, אסורה בהחלט. לקבלת אישור לשימוש בתוכן צור קשר בדוא"ל [info.testingworld@gmail.com](mailto:info.testingworld@gmail.com)





## יפעת שריפי



במהלך השירות הצבאי צברתי ניסיון בעבודה בפרוייקט ERP, שנתן לי הזדמנות להיחשף להכל מהכל - פרודקט, בדיקות, ניהול משתמשים, תמיכה טכנית, הדרכות ובעצם, מה לא.

אחרי השירות התמקדתי בתחום הפרודקט - אבל לקח לי שנה להבין שזה פשוט לא בשבילי. ניסיתי למצוא את עצמי מחדש, למצוא את המקום שמתאים לי - והיה זה בעלי היקר (ינון) שהציע את תחום הבדיקות שהתאים לי כמו כפפה ליד.

האופי שלי מותאם באופן מוחלט לבדיקות. כשחשבתי על המעבר לתחום הבדיקות ודיברתי עם אנשים היו הרבה שהגיבו "איך לא חשבנו על זה קודם!".



## כיצד את מניעה (Motivate) את הבודקים ומה עוד היית רוצה לעשות?

המוטיבציה שלנו מגיעה מהתוכניות שיש לנו לצוות. אנחנו מתכננות לגדול ולהרחיב את הפעילות של הצוות לתחומים נוספים ולהמשיך ללמוד טכנולוגיות וכלים חדשים שיאפשרו לנו לעבוד בצורה טובה ומהירה יותר. בכל פעם שדיברנו השנה על התוכניות לשנה הקרובה אני אמרתי שאנחנו הולכות להשתלט על העולם ובדרך, נשתלט על איכות כל החלקים של המוצר הכולל ב-Fyber.

## כיצד הנכם פועלים להעשרת הידע של הבודקים? (הן מבחינה מתודולוגית והן מבחינה טכנית)

אני מקפידה להקדיש זמן בכל ספרינט ללמידה. בכל יום יש שעה שכל אחת בצוות יכולה לשים בצד את העבודה השוטפת בכדי ללמוד כל נושא שמעניין אותה - ואני מעודדת את עצמי ואת הצוות להשתמש בזמן הזה היטב. נושאים רלוונטיים ושימושיים לכל הצוות עוברים הלאה, ואנחנו מקפידות להעשיר אחת את השניה עם נושאים - גם בליבת העבודה, וגם מעבר.

אנחנו מאוד מאמינים בהעשרת הידע של חברי הצוות, אם זה בהשתתפות בוויבנרים ומיטאפים, קורסי העשרה ושיתוף סרטונים וחומרי קריאה בתחום. יש בינינו המון שיתוף מידע, תמיד שואפים להיות בקדמת הטכנולוגיה, משתדלים כל הזמן להתעדכן ולבדוק את הכלים החדשים ביותר ולבחון האם הם מתאימים לנו. כתרבות של החברה באופן כללי, אנחנו מקפידים על זה מאוד. המוצר שלנו, בין היתר, מציע בדיקת את זה, לאפשר לעובדים לצבור חוויות חדשות בתוך הארגון, ללמוד כישורים חדשים, להתפתח, ולשאוף תמיד להיות בתנועה למעלה. זה מה שאנחנו מציעים גם ללקוחות שלנו לעשות.



## במה עוסקת הקבוצה וכיצד היא בנויה?

ב-Fyber יש לנו כמה צוותי QA. כל צוות אחראי על חלק אחר בכל אחד ממגוון המוצרים שלנו. אני ראש הצוות של בדיקות פלטפורמת המשתמשים שלנו. הצוות שלנו מורכב מ-4 נשים מדהימות במגוון שלבים בחיים (החל מסטודנטית ועד לאמא ל-2 ילדים בגיל יסודי) וכל אחת מאיתנו מביאה משהו שונה וייחודי לצוות שרק הולך ומשתבח עם הזמן.

## איך נראה יום העבודה שלך?

יום העבודה שלנו משתנה כל הזמן. יש ימים שאנחנו עסוקות בבדיקות של פיצ'רים חדשים, יש ימים שאנחנו עסוקות בכתיבת בדיקות, יש ימים המלאים בשיחות ודיונים וישנם ימים שאנחנו לגמרי עמוסות בבדיקות גרסיה. העבודה ממשיכה להיות דינמית, עם עבודה צמודה בהתאם ליציאת פיצ'רים וצרכים יום יומיים - כך שניתן לומר כי אף פעם לא משעמם אצלנו.

"אנחנו מאוד מאמינים בהעשרת הידע של חברי הצוות, אם זה בהשתתפות בוויבנרים ומיטאפים, קורסי העשרה ושיתוף סרטונים וחומרי קריאה בתחום"

## מה האתגרים שלכם בתחום הבדיקות וכיצד אתם מתגברים עליהם, האם נראה כי אלו ישתנו בעתיד?

כמעט כל האתגרים שלנו הם אתגרים שאנחנו בוחרות להתמודד איתם, ישר ולעניין. המבחן שלי הוא להאמין בצוות שלנו, להאמין שאנחנו יכולות להתמודד עם כל מה שעומד מולנו - והבנות בצוות זורמות איתי גם אם הן לא בטוחות. הן מגלות בדיעבד שאני לא מאמינה בהן סתם. ככל שאנחנו משתפרות וצוברות ניסיון בתחומים מסוימים, אלו הופכים למוכרים יותר ומורכבים פחות - האתגר הבא מופיע ואנחנו משתפרות כל הזמן כדי לתת מענה אופטימלי לחברה. לא חסרות הזדמנויות להכיר עוד זוויות במגוון המוצרים של Fyber וזה עולם שלא מפסיק לאתגר אותנו.

## מהם האתגרים הייחודיים של קבוצת הבדיקות שלכם - וכיצד אתם מתמודדים אתם?

כרגע, מאחר ואנחנו הצוות שאחראי בעיקר על ה-Full Stack Web UI, האתגרים הייחודיים שלנו הם התמודדות עם סטטים אוטומטיים לא יציבים שעדיין קיימים לנו בקוד. ה-code base הנוכחי הוא הפעם השניה שכתבנו את כל הסטטים מההתחלה, אחרי שלמדנו המון מטעויות שעשינו בסבב הכתיבה הראשון. אחרי שקראתי את הספר Clean Code (אגב, מומלץ לכל מי שכותב קוד), אנחנו משתדלות לעקוב אחרי הלקחים שהוא מלמד שם ולפעול לפיהם.





## פספורט קבוצתי – **Fyber**

חברת **Fyber** מספקת טכנולוגיה שעוזרת למפתחי אפליקציות ומפרסמים למקסם את ההכנסות שלהם מפרסום במובייל - ומגיעה ליותר ממיליארד משתמשים ייחודיים בחודש בלמעלה מ-180 מדינות. לחברה מאות עובדים במשרדים בישראל, ניו יורק, סן פרנסיסקו, ברלין, לונדון, סיאול ובייג'ין. **Fyber** נרכשה לאחרונה על ידי חברת דיגיטל טורביין האמריקאית, שנסחרת בנאסד"ק. הצטרפות **Fyber** לדיגיטל טורביין ממצב את החברות כפלטפורמת הצמיחה הגדולה ביותר למפתחי אפליקציות ל-**Fyber**.

**סוג המוצר הנבדק:** מוצרי פלטפורמת דפדפן.

**גודל קבוצת הבדיקות:** הצוות כולל 4 נשים. אנו עובדות ולומדות כדי שהמשימות יתחלקו בינינו באופן שווה.

**וותר הבודקים:** בין חצי שנה ועד עשר שנים.

**מבנה הקבוצה:** אנחנו שייכות לקבוצת ה-infra שהיא אחת מבין ארבע קבוצות שמרכיבות את ה-R&D

**סוגי בדיקות:** Compatibility, UI, Functional & Usability

**שיטת עבודה:** סוג של Agile עם ספרינטים של שבועיים.

**כמות המוצרים הנבדקים וקצב שחרור גרסאות:**

כרגע ה-UI שלנו מתחלק לשלושה מוצרים ולאחרונה התחלנו להוציא גרסה פעם בשבוע.

## כיצד הנכם פועלים להעשרת הידע של ראשי הצוותים ומנהלי הקבוצה?

באותה דרך כמו הבודקים שלנו. אני מקפידה בכל תחילת יום, בשניה שאני פותחת את המחשב, להקדיש שעה ללמידה. זה חסום לי בלו"ז בתשע בבוקר ובתשע בערב אני יושבת לשעה ולומדת. לפעמים זה קורס אינטרנטי, לפעמים זה ספר מקצועי, לפעמים זאת הרצאה. העיקר לאתגר את המוח, ללמוד משהו חדש שיכול להעלות לי רעיונות חדשים או ללמד אותי דרכים חדשות לבדוק.

## באילו פעילויות קהילתיות את ועובדיך השתתפתם לאחרונה, ומהן התובנות שלקחתם מהן?

בימים אלו, אני מכינה הרצאה שאעביר במסגרת Meetup שצפוי להתקיים בקרוב. העברתי גרסה מקוצרת של אותה הרצאה בקורס QA של עמותת "צופן" שעובדת לשילוב אנשים מהחברה הערבית בתחום ההייטק.

בתוך **Fyber**, הקמתי קבוצה שכוללת את כל אנשי הבדיקות (והאנשים שבדיקות מעניינות אותם) הקבוצה מוקצת לדיונים, מקום להעלאת שאלות, שיתוף מידע - וגם שיתוף תסכולים מה שלא יהיה.

## אנא שתפי אותנו בכמה מההישגים העיקריים של קבוצת הבדיקות.

כל אחת מאיתנו הביאה איתה הישגים מדהימים. בזכות נגה - התחלנו לתעד את הבדיקות שלנו בצורה מעמיקה יותר (לדוגמא, תיעוד של הטבלאות הרלוונטיות לבדיקה), בזכות ענבל - האוטומציה שלנו קפצה מדרגה בשנה האחרונה ובזכות קרן - כמות הבאגים המוטורפים שלנו קפצה פלאים. אני הבאתי חשיבה אחרת על האוטומציה בעקבות קורס שעברתי לפני כמה חודשים שבו למדתי איך להעריך את הסיכונים בכל גרסה, איך לתכנן את הריצות בצורה יעילה ועל מה לשים את הדגש כשמתכננים את הבדיקות לכל גרסה. באופן כללי, הצוות הופך מצוות של אנשי בדיקות לצוות של אנשי איכות המוצר.

## מהן התובנות שלך לגבי תחום הבדיקות?

אוטומציה זה חשוב. אבל אין אוטומציה בלי האנשים שכותבים אותה, שמתחזקים אותה, שמבינים אותה. אוטומציה היא לא ריצת 100 מטר, היא מרתון. צריך ללמוד, להעמיק, להבין מה אתה רוצה לבדוק והאם ניתן לבדוק את זה לפני שיושבים לכתוב את הקוד.

## מה היית ממליצה לבודק תוכנה שנמצא בתחילת הקריירה שלו?

את אותה המלצה שכתבתי בתשובה הבאה. כמו התובנות בשאלה הקודמת - צריך מההתחלה להעריך את האלמנט לטווח ארוך של העבודה המדהימה הזאת. אוטומציה נמצאת היום בליבת ההתפתחות של התחום וצריך להכיר אותה, ללמוד ולעמיק בה כדי להשיג את התוצאות שרוצים בכדי להתקדם קדימה.

"אוטומציה היא לא ריצת 100 מטר, היא מרתון. צריך ללמוד, להעמיק, להבין מה אתה רוצה לבדוק והאם ניתן לבדוק את זה לפני שיושבים לכתוב את הקוד."

## מה הן ההמלצות שלך לבודקים ומנהלים בתחום?

יש אתר מדהים שנקרא [Ministry of Testing](https://www.ministryoftesting.com/) שאני ממליצה למתחילים, בודקים מנוסים ומנהלים להצטרף אליו וללמוד ממנו, יש שם מידע בכל מיני סוגים (סרטונים, מאמרים, קורסים, וובינרים וכו') על כל נושא שתוכלו לחשוב עליו שקשור לבדיקות תוכנה.





**טל פאר**

בעל ניסיון של יותר מ-20 שנים כבודק ומנהל בדיקות במגוון חברות וטכנולוגיות במודלי פיתוח שונים. כיום טל יועץ ומדריך בדיקות עצמאי.

טל חבר ב-ITCB® וגזבר הארגון העולמי ISTQB®.



בודקים נתפשים לעיתים כאחראים לאיכות התוכנה ולכך שלא יהיו פגמים (defects, bugs) בתוכנה שנמסרת ללקוחות. בתחילת דרכי נשאלתי, עם סיום הבדיקות, האם מצאתי את כל הבאגים במערכת. עניתי בשאלה "כמה באגים הכנסת למערכת?".

כדי שהבדיקות יהיו יעילות ויחשפו את הכשלים והפגמים החשובים ביותר (למשל, הקריטיים ביותר), כמו גם את מספר הפגמים הרב ביותר, על צוות הבדיקות לעבוד לפי מספר עקרונות. הסילבוס של ISTQB® לרמת הבסיס (CTFL) מציג מספר עקרונות מנחים בסיסיים ואלה משותפים לכל סוגי הבחינות. עקרונות אלה הוצגו על ידי מספר אנשים לאורך השנים ו-ISTQB® ריכז אותם לרשימה אחת. מחוץ לסילבוס אפשר ללמוד על עקרונות אלה בספרים כמו [Software Testing Techniques](#) של בוריס בייזר (Beizer) ו-[The Art of Software Testing](#) של גלנפורד מאירס (Myers).

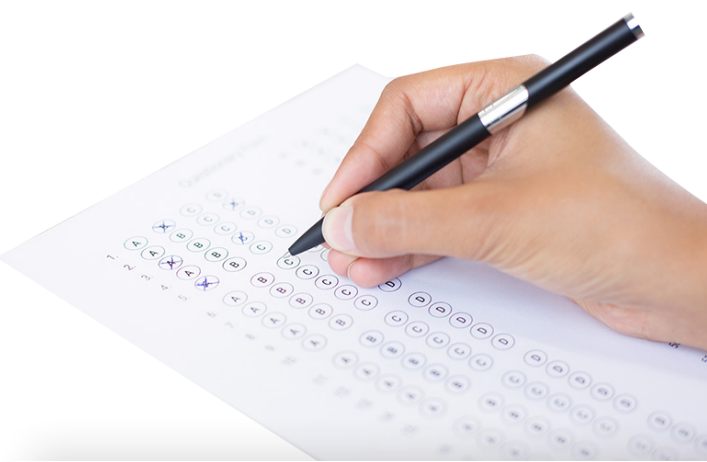
השאלה של היום מציגה איך להשתמש בעקרונות אלה.

(לשם הבהרה אומר שאני בוחר להשתמש במושג פגם, כמו שמופיע בסילבוס בעברית, למילה שבשפת היומיום אנחנו קוראים "באג" מאחר וה"באג" אותו אנו מוצאים בזמן הבדיקות הוא למעשה ה"כשל").

צוות הבדיקות שלך סיים את הרצת סבב הבדיקות האחרון שתוכנן והערכה היא שמצב איכות התוכנה הוא טוב וניתן לשחרר את המוצר. כשהצגת את דו"ח הבדיקות המסכם להנהלת הפרויקט, נשאלת אם קיימים בתוכנה פגמים (defects, bugs) שלא התגלו על ידי הבדיקות.

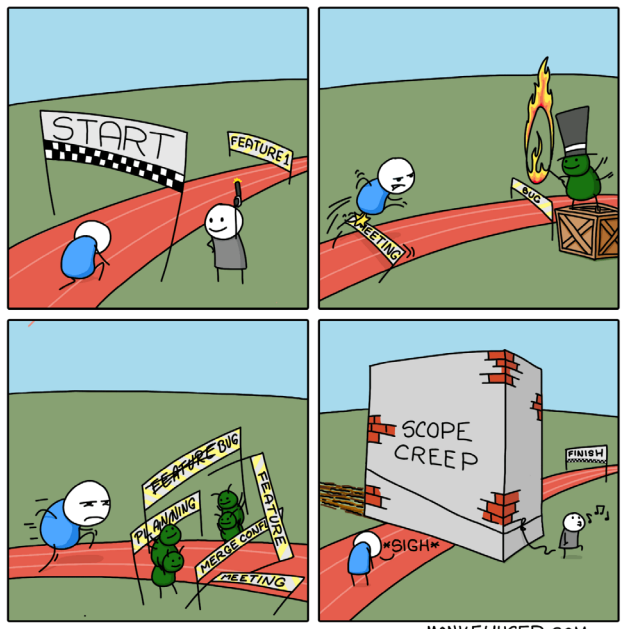
מה תהיה תשובתך, תוך הצדקה באחד משבעת עקרונות הבדיקה?

- א. אין בתוכנה פגמים נוספים מאחר צוות הבדיקות חזר על הבדיקות מספר פעמים ולא גילה פגמים חדשים.
- ב. אין בתוכנה פגמים נוספים מאחר וצוות הבדיקות היה מעורב בתהליך הפיתוח משלב מוקדם.
- ג. קיימים פגמים נוספים שצוות הבדיקות יוכל לגלות אם יתאפשר לו לבצע בדיקות ממצות (exhaustive testing).
- ד. לא ניתן לדעת אם קיימים פגמים נוספים מאחר והבדיקות מגלות פגמים אך לא יכולות להוכיח שאין פגמים נוספים.



**\*לצפייה בתשובה המפורטת -דפדפו לעמוד 14**

**SPRINT**

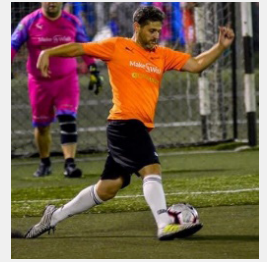






קצת על עצמי...

שמי **אסף סהר**, בן 40, נשוי להדר ואב לשלושה: איתי, דרור וליבי. אני עובד בחברת AppsFlyer כבודק תוכנה שמתמקד בסקויריטי. חולה על כדורגל, אוהד את ריאל מדריד ומשתדל לנסוע פעם בשנה לראות משחק שלהם. אני מאוד משפחתי ואוהב לטייל.  
בעל בלוג: <https://asafsahar25.medium.com>



## איך הכל התחיל

הקריירה שלי בהייטק התחילה בשנת 2007 בחברת openTV כשהייתי סטודנט שנה ב' במדעי המחשב.

בעזרת אחותי טליה שהכירה מישהו שעבד שם והעבירה את קורות החיים שלי. התחלתי במשרת סטודנט במתן תמיכה טכנית ללקוחות החברה במשמרות לילה (התמיכה הייתה בלקוחות מארצות הברית, בחברות אשר השתמשו במערכת שלנו), מספר פעמים בשבוע.

בחופשת הסמסטר בין שנה ב' לשנה ג', המנהלים שלי החליטו שאכנס לתהליכי הבדיקות של המוצר כדי שאבצע אותן בזמן המשמרות וגם הציעו לי להגיע במהלך היום בכדי לעשות חפיפה עם אנשי הבדיקות וללמוד את המקצוע. לדעתי המהלך הזה הגיע כדי שבלילה הם יקבלו ממני תפוקה מירבית אבל גם כי היה להם צורך לעוד ידיים עובדות שיעשו בדיקות. היה מדובר בחברה של בסביבות 30 אנשים כאשר עבדתי שם.

במהלך שנת הלימודים השלישית שלי נוצר מצב שהתחלתי לעבוד יותר כבודק תוכנה ופחות כתומך טכני ונכנסתי יותר לעומק בבדיקות למערכת הדסקטופ של החברה.

בחברת openTV עשיתי את צעדי הראשונים בעולם ההייטק, למדתי מושגים והתחלתי להתרגל ולתרגל עבודה באנגלית בכל הנוגע למסמכים ומיילים. אני זוכר שאז חשבת', למה אני כותב לאיש הזה מייל באנגלית ששינוי דוברי עברית, מה הקטע?

היום אני מבין שזה פשוט יותר כי גם ככה רוב הדברים שאנחנו כותבים יכללו מושגים שפשוט יותר לכתוב באנגלית.

לאחר סיום התואר לא נמצא לי תקן למשרה מלאה ולכן החלטתי להמשיך הלאה בחיפוש אחר משרת חלומותי.

## המשרה המלאה הראשונה שלי בעולם ההייטק

בעזרת חבר מהלימודים שהעביר את קורות החיים שלי לחברת איטריידר שלימים התווסף לה אתר

בינלאומי בשם anyoption, הגעתי לראיון עבודה. התראיינתי אצל מנהל הפיתוח, אחד מהבעלים של החברה ומשאבי אנוש. בעיקר זכורה לי ההתרגשות מכל אחד מהראיונות מאחר וזו הייתה הפעם הראשונה שלי, כג'וניור, שרק סיים את התואר הראשון. מאוד רציתי להתקבל.

לאחר שהתקבלתי התחלתי כבודק ידני ונחשפתי לבדיקות דפדפנים לראשונה ברמה מעמיקה, עבודה עם מסדי נתונים וכתובת שאילות.

זכור לי הפיצ'ר הראשון שבדקתי, זה היה דף באתר שלנו שהציג היסטוריית עסקאות עם תאריכים שהיו משתנים לפי אזורי זמן, ולא היה ברור לי אז אם מדובר לפי שעון זמן של המחשב שלי או של השרת שעליו רצה התוכנה. אני זוכר שהיו לי הרבה שאלות למפתח שכתב את זה והשאלות שלי אליו נתנו לי את ההרגשה שהוא כזה חושב, הנה עוד אחד שרק סיים לימודים, רוצה להוכיח שהוא יודע משהו ואין לו מושג בכלום.



## היעד הבא...

לאחר תקופה של 4 שנים בחברת איטריידר החלטתי לעבור ליעד הבא, עברתי לחברת Neogames, שם התחלתי כבודק תוכנה ידני ולא כראש צוות, למרות שבאתי מלכתחילה לראיון העבודה לתפקיד ראש צוות. הסכמתי

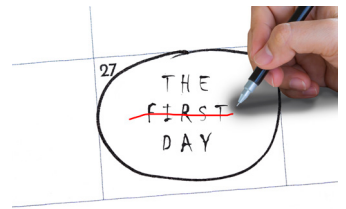
לזה בגלל הרושם הטוב שקיבלתי מהמקום שאני אוכל ללמוד בו עוד הרבה דברים כבודק תוכנה, להכיר את המערכת טוב ללא הלחץ של התפקיד הניהולי ואז להתקדם משם. דבר אשר הוכיח עצמו בצעד חכם ונכון. לאחר תקופה של מספר חודשים מוניתי לראש צוות. בתקופות מסוימות ניהלתי עד 6 עובדים. למדתי שם הרבה, למשל - איך להיות בודק יותר טוב, מנהל יותר טוב וגם מבחינת נושאים אחרים כמו תהליכי פיתוח מוצר, עבודה מול לקוחות ועוד.

## אוטומציה - כל השנים רציתי לכתוב קוד

לאחר תקופה של בין 3 ל-4 שנים נפתחה בפני האפשרות להצטרף למחלקת TechOps כמפתח

אוטומציה בצוות קיים. כמובן שלקחתי את ההזדמנות הזו והצטרפתי לצוות. למדתי הכל מאפס. למדתי עקרונות פיתוח בעולם האמיתי ולא בהכרח דברים שלמדתי בתואר מספרי שנים. למדתי מה זה Page Object model ועוד. למדתי מה זה Page Object model ועוד.

זה המקום להודות למנהל המחלקה וראש הצוות שהיה לי אותה תקופה שעזר לי מאוד ללמוד C#, סלניום ואת התשתית לאוטומציה שפותחה In House. התפקיד שלנו היה לקחת טסטים ידניים של בדיקות גרסיה אשר נכתבו על ידי מחלקת הבדיקות ולהמיר אותם לטסטים בסלניום שירוצו בזמן העלאות גירסה. הטסטים רצו על פרודקשן במובייל ועל דפדפנים כאשר גם את התשתית שעליהן רצה האוטומציה אנחנו ניהלנו. בעזרת הידע שרכשתי במשך השנים





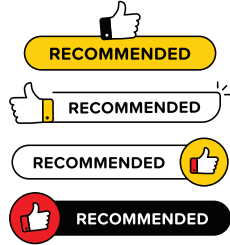
לעומק איך רשתות אינטרנט עובדות (זה מאוד יעזור ביכולות להבין כיצד דברים עובדים ואז לנסות לעקוף אותם) וגם ללמוד יותר על עולמות הענן.

## המלצותיי

קוראים של מגזין הבדיקות, כנראה רובכם מבצעים בדיקות או רוצים להתחיל לעסוק בזה, ההמלצה שאני יכול לתת לכם זה לדחוף לכיוון של לשלב בדיקות אבטחה, לבקש לצאת לקורס או ללמוד עצמאי (יש המון חומר באינטרנט). כחלק מהכנת תכנית הבדיקות תוסיפו פרק עבור בדיקות אבטחה ותתחשבו בזה בהערכת הזמנים שאתם נותנים. דברו

עם אנשי הסקויריטי אם יש לכם היכן שאתם עובדים, תתעניינו מה הפגיעות הכי נפוצה אצלכם בארגון, תנסו לקבל מהם כמה שיותר מידע, תראו אם אתם יכולים להתחיל ללמוד איך מוצאים אותה ואז להכין בדיקות מתאימות. תחושת הסיפוק שתהיה לכם שתמצאו את באג האבטחה הראשון שלכם תהיה שווה את המאמץ!

תודה רבה שהקדשתם מזמנכם לקרוא את המאמר



במחלקת הבדיקות על המערכת הצלחתי ביחד עם ראש הצוות להכניס **API Requests** ששימשו כתנאי מקדים לבדיקות במקום לבצע אותם בסלניום, דבר אשר תרם ליציבות הטסטים וזמני ריצה קצרים יותר. (לדוגמה לרשום משתמש חדש למערכת או להפקיד כסף לחשבון).

לאחר כשנה התפנה תפקיד של ראש צוות ואני מונית אליו. קיבלתי על עצמי אתגר גדול להכיר את תשתית האוטומציה יותר לעומק ולשפר אותה היכן שצריך, לבדוק חלופות לדרך הרצת הטסטים, כמו למשל סלניום גריד, Kubernetes, Azure ועוד.

ניהול התשתית היה מאתגר במיוחד, זו הייתה תשתית של מכונות וירטואליות שהיה צריך לנהל אותן מבחינת גרסאות דפדפן, מתי לעדכן ואיך. בנוסף, עבדנו עם TFS של מיקרוסופט והיינו צריכים מכוונת אחת ששימשה מאין מנהלת תהליכים ובאחרות היו איג'נטים שכל מה זמן היה צריך לעדכן אותם ואז זה היה גורם לפעמים לבעיות. כדי לפתור חלק מהבעיות האלו למדתי לעבוד עם Powershell כדי להריץ מספר דברים אוטומטיים על המכונות כדי לתחזק אותן. לכן פנינו אחר כך לנסות ולמצוא פתרונות אחרים שציינתי למעלה. שם בכלל נכנסתי לאתגרים חדשים שאם אאלץ לכתוב עליהם פה אז לא נסיים 😊

## בדיקות סקויריטי – נשמע מעניין



לאחר כ-3 שנים בצוות האוטומציה ועוד 4 שנים בערך במחלקת הבדיקות של Neogames הגיעה אלי הצעה מ-**AppsFlyer** לתפקיד חדש יחסית בעולם הבדיקות בשם QA Security.

אני זוכר שאמרתי להם בשיחות הראשונות שאני לא יודע כלום בהקשר לאבטחת מידע ולא עשיתי בדיקות כאלו מעולם.

מטרת העל של התפקיד היא להיות האיש המקשר בין מחלקת אבטחת המידע לבין מחלקת הבדיקות ולהעלות את המודעות לסקויריטי. בתפקיד הזה למדתי (ועוד לומד) כיצד עושים בדיקות אבטחה ואני מעביר את הידע הזה לצוותים במפגשים לכל מחלקת הבדיקות ובמפגשים של אחד על אחד עם הבודקים על פיצ'רים ספציפיים.

בנוסף פיתחתי חבילה (Package) בשפת פייתון. בשפה זו מורידים הרבה חבילות לפרויקט שלך כדי להשתמש בהם ולקודד בעזרת הפונקציות שאותן חבילות מספקות את מה שאתה צריך. החבילה שפיתחתי מבצעת בדיקות לסוגי פגיעות שונות ב-Web שהצוותים משתמשים בה ומרצים כל אחד על ה-Services שהם בודקים. המערכת מבצעת את הבדיקות ע"י **API Requests** ובעתיד יתכן והחבילה תשלב גם בדיקות ממשק משתמש.

האתגר כאן הוא גדול, ברמה האישית הייתי צריך ללמוד על עולם בדיקות ה-Web Security, ללמוד שפת תכנות חדשה - פייתון, לעבור למקום עבודה חדש אחרי 7 שנים במקום אחד (ואצין גם שחודש וחצי אחרי שהתחלתי לעבוד שם פרצה הקורונה מה שהוסיף לאתגר), לעבוד בחברה בסדר גודל כזה של כ-1300 אנשים ויותר מ-45 בודקים ובודקות תוכנה ועם הרבה משאבים.

חלק עיקרי מהתפקיד זה להנחיל שינוי בחשיבה אצל בודקי התוכנה שבדיקות אבטחה גם צריכות להיות חלק מתוכנית הבדיקות שלנו כמו בדיקות פונקציונאליות או עומסים.

מה שהדהים אותי כשהתחלתי לבצע בדיקות אבטחה זה שחלק מהבדיקות שעשיתי בעברי היו ממש קרובות לדרך שבה אפשר לבצע בדיקות סקויריטי מסוימות למשל כמו בבדיקות ערכים של API Request. אם חושבים על זה, כשבוחנים קריאות לשרת בסניפר כלשהו, למה לא לנסות לשים ערכים שאולי בעזרתם ניתן להגיע לנתונים של משתמש אחר? היום זה מרגיש לי שזה היה שם, ממש מתחת לאף.

בשנים הקרובות אני מתכוון להתמקצע עוד בבדיקות אבטחה, זה תחום עצום, כרגע אני רק מתמקד בחלק אחד שזה בדיקת אפליקציות ווביות וגם בו יש המון מה ללמוד. תחשבו שכל מה שמחובר לאינטרנט ניתן לפרוץ אליו ולעשות בדיקות, בין אם זה מתכונות, מוצרי חשמל, בתים חכמים ועוד. הייתי גם רוצה ללמוד יותר

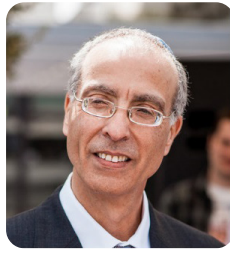






## על הקצה

הטור הקודם (תוכנית החלוקה) עסק בטכניקת הבדיקה של מחלקות שקילות. הפעם אתמקד בטכניקה הקשורה לה: בדיקת ערכי גבול, וגם כאן יתברר שהחיים יותר מסובכים מאשר התרגילים שפותרים בקורס בדיקות.



### מיכאל שטאל

ארכיטקט בדיקות תוכנה באינטל, ישראל עוסק בעיקר בבדיקות מערכות משובצות מחשב. במסגרת תפקידו, מיכאל מגדיר שיטות בדיקה ומתודולוגיות עבודה, עוסק בהדרכה ולפעמים אפילו מרשים לו לבדוק משהו (שזה הכי כיף). מיכאל מציג תכופות בכנסים בארץ ובחו"ל ומלמד בדיקות תוכנה בפקולטה למדעי המחשב באוניברסיטה העברית. ניתן לראות חלק מהמצגות והמאמרים שלו באתר [www.testprincipia.com](http://www.testprincipia.com)



שקילות על ידי משפטי תנאי (if-else) ואז קל לזהות את מקרי הגבול של התנאים. כיוון שממילא צריך לבחור נציגים מכל מחלקה לצורך כיסוי כל המחלקות, וכיוון שהחיים מראים שיש שכיחות טעויות בקצוות של ערכי הקלט – אז הגיוני לבחור את ערכי הגבול בתור הנציגים של מחלקות השקילות. גם כיסינו את המחלקה, וגם בחרנו ערכי קלט שיש יותר סיכוי שימצאו באג.

### "יש קשר הדוק בין מחלקות שקילות וערכי גבול"



הצד הפחות חביב של הקשר למחלקות השקילות הוא שהאתגרים הקיימים בחלוקה למחלקות שקילות משפיעים באופן ישיר על נכונות בחירת ערכי הגבול: אם טעינו בחלוקה למחלקות, הטעות תיגרר להגדרת ערכי הגבול. זה גם מעלה את מספר הבדיקות הנדרש: אם עבור כיסוי מחלקות שקילות הסתפקנו בנציג יחיד מכל מחלקה ולכן מקרה בדיקה יחיד, לכיסוי גבולות של טווח מסויים (למשל... דוגמא מקורית מאוד: ערכים מ-1 עד 10), צריך כבר ארבע מקרי בדיקה (0,1,10,11) וזה כשמתמשים בשני ערכי גבול. אם משתמשים בשלושה ערכי גבול אז כבר מדובר בששה טסטים (0,1,2,9,10,11).

### תזכורת: בדיקות ערכי גבול

#### REMINDER



טכניקת הבדיקה של "ניתוח ערכי גבול" מבוססת על כך שסיבות נפוצות לבאגים הן תופעות שקשורות לגבולות. ישנם כמה מקרים שבהם אפשר לדבר על "גבול": המקרה הפשוט ביותר הוא כאשר התוכנה מצפה לקבל קלט בטווח מסויים. למשל - שדה החדש בתאריך: הציפיה שיהיה בין 1 ל-12. קצת יותר מסובך: כשת-טווח של הקלט קובע את התנהגות התוכנה: מגיל 0 עד 2 לא משלמים על כרטיס טיסה; מגיל 2 עד 12 משלמים 75%, ומגיל 12 עד 120 - מחיר מלא.

במקרים אלה ישנן מספר טעויות אפשריות:

- (א) המפתחים לא הבינו נכון את הדרישה (למשל: חברת התעופה "מבינה" שגיל 0 עד 2 כולל ילדים שעברו את גיל 1, אבל אינם עדיין בני שנתיים; המפתחים הבינו שזה כולל ילדים שעברו את גיל שנתיים אבל אינם עדיין בני שלוש)
- (ב) המפתחים עשו טעות בקוד. במקום לכתוב " $1 >=$ " הם כתבו " $2 >=$ "
- (ג) המפתחים לא ציפו שיש בכלל אפשרות לקבל קלט מעבר לטווח המצופה. אם נניח שהטיפול בערך החדש הוא על ידי פקודת switch (או case בשפות אחרות), אז אין קוד default שמטפל במקרים שבהם הקלט אינו חוקי (למשל 13).

מכאן ההמלצה לתכנן בדיקות שהקלט שלהן מכיל ערכי גבול. סביר כל גבול נריץ ערך אחד שהוא בטווח, אבל האחרון שעדיין שם (ילדים בני 0 ו-1; חודשים 1 ו-12) וקלט ש"בדיק" כבר לא בטווח (ילדים בני 2; חודשים 0 ו-13).

### זה כל כך פשוט!



דוגמא פשוטה ונפוצה ביותר בכל מקום שבו מוסברת הטכניקה, היא עבור קלט שיכול לקבל ערכים בין 1 ל-10. ערכי הגבול הם 0, 1 עבור הגבול התחתון, ו-10, 11 עבור הגבול העליון. כבר כאן יש הצגה מאוד פשטנית של הטכניקה. קודם כל, לא תמיד הקלט הוא במספרים שלמים; מה הגבולות אז? לפעמים הקלט הוא מחרוזת; מה הגבולות? כמו כן, יש מקרים שבהם ערכים מעל 10 או מתחת ל-1 אינם ערכים שגויים, אלא ערכים שעבורם המערכת מתנהגת אחרת. איך לקבוע את ערכי הגבול במקרים אלה? ישנם גם מקרים שבהם לא מדובר במשפט תנאי של "אם הערך גדול מ-X אז..." אלא הגדרת תנאי סיום ללולאות של for או while, ושם המצב יותר מסובך. תוסיפו על זה את האפשרות להגדיר שני ערכי גבול או שלושה ערכי גבול, וכבר מובן שיש מקום להיכנס קצת יותר לעומק.

### ערכי גבול ומחלקות שקילות

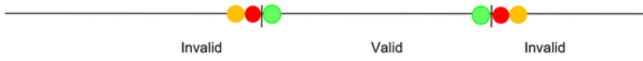
נתחיל בזה שיש קשר הדוק בין מחלקות שקילות וערכי גבול. טכניקת הבדיקה של חלוקה למחלקות שקילות מציעה לבצע בדיקות עם ערכי קלט שמייצגים את מחלקות השקילות - לפחות נציג אחד מכל מחלקה. בדרך כלל הקוד מחלק את מרחב הקלטים למחלקות





"אם טעינו בחלוקה למחלקות, הטעות תיגרר להגדרת ערכי הגבול"

זכרו גם שאת הערכים שאינם על הגבול אפשר לשים צמוד לערך הגבול:



קיבלנו משהו שאכן נראה כמו שלוש ערכים לכל גבול. מ.ש.ל.

יש רק בעיה קטנה: על פי התיאוריה של שלושה ערכי גבול מדובר בערך שעל הגבול, ושני ערכים משני צידי. כלומר, בעצם צריך להסביר בחירת בדיקות כזאת:



יתכן שההסבר נעוץ בדרך שבה מתארים את הבעיה. הרי זה לא באמת כמו הציונים למעלה. אם הטווח המותר הוא "בין 1 ל-10" אז התיאור הנכון עבור בדיקות עם שני ערכי גבול הוא שהערך התקף יושב ממש "על" הגבול:



זה קצת מציק בעין. נראה איכשהו לא סימטרי. ומכאן הדרך סלולה למשהו שנראה יותר טוב (אם כי לא בטוח כלל שנותן בדיקות חזקות יותר):



לא כולם ביחד!

העולם כידוע יותר מסובך מקלט אחד בטווח של 1 עד 10. בהרבה מקרים הקלט מכיל יותר ממשנתה אחד ולכל משנתה יש, מה לעשות, ערכי גבול. איך נממש את הטכניקה של בדיקות ערכי גבול במצב זה?



לצורך ההמשך, נניח שאנו מפתחים תוכנה שמזהה עצמים בתוך תמונה. התוכנה משתמשת בחומרה שעושה את רוב העיבוד, ועל מנת שהחומרה תהיה פשוטה יותר, הוחלט שהתמונה שתישלח לחומרה תהיה תמיד באותו גודל: 1500x1000 פיקסלים. התוכנה עצמה מקבלת תמונות בגדלים מ-150x100 ועד 9000x6000 פיקסלים, וכל תמונה שתקבל תעבור המרה לגודל 1500x1000 לפני שתועבר לחומרה. תמונות שבהן היחס בין הגובה לרוחב אינו 1:1.5, ידחו כלא חוקיות.

נתחיל בזה שעכשיו יש לנו ארבעה פרמטרים שאפשר לנתח אותם מבחינת מקרי גבול:

גובה התמונה: 100 עד 6000

רוחב התמונה: 150 עד 9000

שטח התמונה: 150x100 עד 9000x6000

יחס גובה/רוחב: 1:1.5

מפתה להחליט שנבדוק את כל ערכי הגבול יחד:

גבול עליון	גבול תחתון
9001x6001	149x99

שניים או שלושה (רמז: ארבע!)



יש סיבה טובה למה צריך יותר משני מקרי בדיקה כדי לכסות כל גבול. ניקח דוגמה של תוכנה למכירת ביטוח חיים<sup>1</sup>. התוכנה מממשת כלל פשוט: יש אישור לבטח מי שמתחת לגיל 99. אפשר לנחש בוודאות גבוהה שאיפשהו בקוד יהיו שורות שיממשו משהו בסגנון הבא:

If age <= 99 then Accept; Else Reject

אם נשתמש בבדיקות ערכי גבול עם שני ערכים, נריץ שתי בדיקות:

גיל	תוצאה צפויה
99	Accept
100	Reject

אבל מה אם המפתח גגה, ובמקום ">=" כתב "<" ברור שהתוצאות בפועל יהיו הפוכות לתוצאות הצפויות, הבאג ימצא, ואנו הבודקים, נסמן עוד חריץ על קת הרובה. אבל האם, בעזרת שתי בדיקות, נוכל לתפוס את כל השגיאות הלוגיות האפשריות? במאמר המקורי יש טבלה עם כל האפשרויות, וממנה מתברר שיש טעות אחת שלא ניתן לתפוס על ידי שתי בדיקות בלבד:

If age == 99 then Accept; Else Reject;

על מנת "לתפוס" גם את המקרה הזה, צריך להריץ עוד ערך מתוך מחלקת השקילות התקפה, שאינו ממש על הגבול. כל ערך יעשה את העבודה, אז נניח שבחרנו בערך 98.

אבל מה אם הקוד מומש ככה:

If age < 100 then Accept; Else Reject;

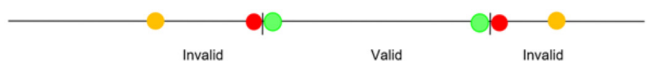
כמעט כל הטעויות הלוגיות יתפסו על ידי הרצת הערכים 98, 99, 100. חוץ מהטעות הבאה:

If age < > 100 then Accept; Else Reject;

בשביל לתפוס טעות זו, צריך להריץ ערך נוסף ממחלקת השקילות שאינה תקפה, שאינו על הגבול. גם כאן כל ערך ייתן את אותה תוצאה, ואם כך אפשר לבחור את 101.

אז הסברנו למה כדאי לבדוק ערכי גבול עם 4 ערכים. גם ההצדקה לבדיקות עם שני ערכים ברורה: זה תופס את רוב הטעויות, וודאי את ה"הגיוניות" שבהן (תסכימו שהמקרים שאינם נתפסים בעזרת שני ערכי גבול הם, איך נאמר... גבוליים). לעומת זאת, מאיפה ההמלצה לשלוש ערכי גבול... לא לגמרי ברור. אני אנסה להסביר את זה בניפנופי ידיים נמרצים, אבל מודע לכך שזה הסבר חלש.

בהרבה מקרים יש למחלקת שקילות שני גבולות: תחתון ועליון:



על פי העקרון של ארבע בדיקות בכל גבול, נריץ: ערך תקף על הגבול (יורק); ערך לא-תקף על הגבול (אדום); ערך לא-תקף שאינו על הגבול (כתום). מה עם הערך הרביעי (ערך תקף שאינו על הגבול)? מסתבר שיש לנו אותו: עבור הגבול התחתון, הערך התקף של הגבול העליון הוא "ערך תקף שאינו על הגבול", וכנ"ל לגבי הגבול העליון.

<sup>1</sup> דוגמה זו והסבר שלה לקוחים ממאמר שהתפרסם במגזין Testing Experience בשנת 2008: The Boundary Value Fallacy, René Tuinhout, Testing Experience Magazine, Sep 2008 [https://www.istqb.org/images/Articles/konushin\\_personnel%20evaluation%20based%20on%20tests.pdf](https://www.istqb.org/images/Articles/konushin_personnel%20evaluation%20based%20on%20tests.pdf)





התמונה. לעומת זאת, על פי הגדרה זו, נוצר לנו פרמטר נוסף (גודל האובייקט) שגם עליו צריך לבצע ניתוח ערכי גבול, ולבדוק בהתאם. נשמע קל, אבל הגדרות מסוג זה הן מאוד קשות, כי הרי זה לא רק גודל העצם; זה הזווית בה הוא צולם; הקונטרסט בינו ובין הרקע; חדות הצילום; וכו' וכו'. לעיתים בלתי אפשרי להגדיר ממש את הגבול, אלא הגדרות הרבה יותר מעורפלות. אז מריצים את התוכנה עם הרבה מקרים, מסתכלים על התוצאה ומחליטים אם היא "טובה מספיק".

יחס הרחבגובה בתמונה גם הוא מקרה מעניין שכן מדובר בפרמטר עם ערך שאינו שלם. מה מקרה הגבול? האם תמונה של  $6000 \times 8999$  (יחס: 1:1.4998333) נחשבת מספיק קרובה ל-1:1.5? אם לא נגדיר את הדיוק שאנו דורשים (למשל:  $\pm 0.002$ ) הרי שהתוצאה תהיה תלויה במערכת ההפעלה ובחומרה עליה רצה התוכנה - כי הם קובעים את רמת הדיוק בייצוג מספרים ממשיים. זה יכול להיות בעייתי אם זו תוכנה שתותקן במערכות שונות ומגוונות, או סביר לגמרי אם התוכנה מיועדת למימוש מאוד ספציפי. בכל מקרה, עלינו כבודקים להבין מהו הדיוק הנדרש, על מנת להגדיר את מקרי הגבול, ומשם הדרך למקרי בדיקה קלה.

**לסיכום**, ישנם מקרים (רבים) שבהם בדיקות מקרי גבול ברורות ומיידיות. אבל יש גם לא מעט מקרים שבהם העקרון לא ממש עובד, או קשה ליישם אותו. במקרים אלה, העקרון של "מקרה גבול" עוזר לנו לחשוב על בדיקות שכדאי להריץ, גם אם הם לא ממשות את הטכניקה לפי הספר. רוח הדברים של הטכניקה מנחה אותנו לחפש מתי התוכנה עוד מתפקדת ומתי מרימה ידיים, ולו גם ברמה אבסטרקטית של חיפוש מקרים ש"כמעט נכשלים" ומקרים ש"כמעט עוברים".



9000×6000	150×100
8999×5999	151×101

אבל יש סיבה טובה למה לא: הקוד. מן הסתם, בדיקת גודל התמונה מתבצעת מיד בהתחלה. משהו בסגנון:

If (height < 100 or height > 6000 or width < 150 or width > 9000) then Error

### " רוח הדברים של הטכניקה מנחה אותנו לחפש מתי התוכנה עוד מתפקדת ומתי מרימה ידיים "

כשקוד זה עובר קומפילציה, התוצאה מכילה Short Circuiting. זהו אופטימיזציה שהקומפילטר מבצע על מנת לחסוך זמן עיבוד בשעת ההרצה. הבה ננתח: כאשר הגובה קטן מ-100, כבר התנאי הראשון מתקיים. כיוון ששאר התנאים הם "או" (OR), אין שום טעם להמשיך ולבדוק את שאר התנאים. ברור שהתוצאה הסופית של כל הביטוי הלוגי תהיה "אמת", ולכן אפשר לנטוש את הביצוע כבר אחרי שהתנאי הראשון מתקיים. מבחינה מעשית זה אומר שאם נשתמש בתמונה בגודל  $149 \times 99$  על מנת לבדוק את הגבול התחתון, גם הקוד הזה ייתן את התוצאה הצפויה:

If (height < 100 or height > 6000 or width > 9000) then Error

ולמעשה המקרה של  $width > 150$  כלל לא נבדק.

ההמלצה לגבי בדיקות מקרי גבול כאשר יש יותר ממשתנה אחד היא לבדוק כל משתנה בנפרד (להציב מקרי קצה למשתנה זה) תוך שאנו מציבים ערכים נומינליים (שאינם בקצוות) לכל המשתנים האחרים. מעבר לזה, מומלץ להריץ בתור התחלה מקרה שבו כל המשתנים בערכים נומינליים. אם בדיקה זו תעבור, ואחרות לא, זו אינדיקציה מצויינת שהבעיה היא בגלל מקרה קצה. אם גם בדיקה זו לא תעבור, אז כנראה אף בדיקה לא תעבור כי לפחות אחד המשתנים מטופל לגמרי לא נכון. בהנחה שיש ח משתנים, שנבחר לבדוק עם שני ערכי גבול, ושלכל משתנה יש גבול עליון ותחתון, מדובר ב-4n בדיקות, ועוד בדיקה אחת שבה כל המשתנים בערכים נומינליים.

כיוון שגם שטח התמונה הוא פרמטר, הרי יש גם הצדקה להריץ את התמונה הגדולה ביותר שאפשר לעבד בתוכנה שלנו:  $9000 \times 6000$ . לעומת זאת, אין טעם לנסות תמונה בגודל גדול יותר (למשל  $6001 \times 9000$ ) כי ברור מבדיקות מקרי הגבול של גובה ורוחב שקלט זה ידחה. גם אין משמעות לבדוק תמונה בגודל  $6000 \times 8999$  או  $9000 \times 5999$ , שהרי מטרת הבדיקה עם תמונה בעלת שטח מקסימלי היא לבדוק שהמערכת מקצה מספיק משאבים לקלוט כמות כזו של נתונים (54 מיליון פיקסלים). אם התמונה הגדולה ביותר מטופלת היטב, הרי שיש במערכת מספיק משאבים לכך, וודאי שיש מספיק לתמונות קטנות יותר.

### גבולות מעורפלים

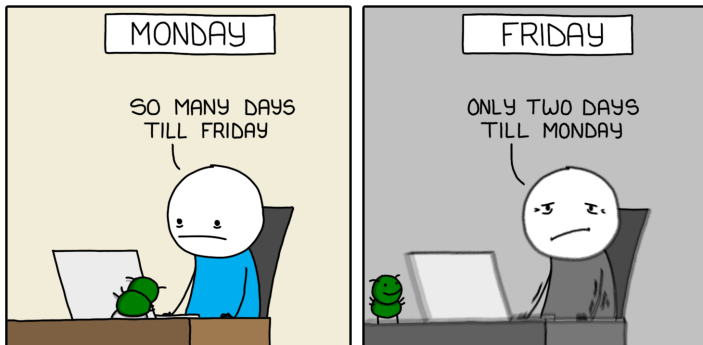
נשארנו עם שני פרמטרי קצה: התמונה הקטנה ביותר, והיחס גובה/רוחב.



איזה מקרה גבול מייצג את התמונה הקטנה ביותר? זה לא חד משמעי כלל, ויתכן שהתמונה הקטנה ביותר אינה מקרה גבול לשום דבר. בדוגמה שלנו התוכנה מזהה עצמים בתמונה. נניח שהתוכנה מזהה עצמים עם גודל של לפחות  $150 \times 150$  פיקסלים. במקרה כזה, תמונה של  $100 \times 150$  כנראה ממש לא מעניינת אלא רק כמקרה בדיקה שלילי להראות שכשאף חפץ לא מזהה - או שבתמונה יש רק חלק מעצם - התוכנה לא קורסת. אם הגודל המינימלי של עצמים הוא  $98 \times 98$  פיקסלים, אז יש (כמעט במקרה) משמעות לגודל המינימלי של

### LOADING WEEKEND

MONKEYUSER.COM





## כבר לא סנונית

אם בעבר, מראה אישה חרדית היוצאת מבניין הייטק, נחשב למחזה נדיר, אולי אפילו תמוה, הרי שכיום נוכל למצוא לא מעט כמוה מועסקות בכל חברות ההייטק המובילות בישראל.

מבלי ששמנו לב, התחוללה לה מהפכה שקטה בעולם התעסוקה החרדי, כאשר שיעור הנשים החרדיות העובדות בהייטק גדל מראשית העשור פי 2.5. היום כ-10% מהנשים החרדיות עובדות בהייטק (!) ומספרן עולה משנה לשנה. מה שהתחיל כיחידות פורצות דרך הפך לתופעה שכיחה וקצת אין שום ספק, העובדות החרדיות כאן כדי להישאר.

## נעים להכיר, בודקות תוכנה הייטקיסטיות חרדיות



אז מי אנחנו?

נשים צעירות, רובנו בגילאי 24-19, מלאות אמיביציה ומרץ געורים. כמקובל במגזר אנחנו מקימות משפחה בגיל מוקדם ויוצאות לעבוד למחייתנו. אנו בוחרות ליטול על עצמנו את עול הפרנסה, בכדי שהגברים יוכלו להקדיש את חייהם ללימוד התורה והתעלות רוחנית, שהם ערך עליון בהשקפת עולמנו.

אחרי התיכון אנו ממשיכות לשנתיים-שלוש של לימודי תעודה מקצועיים בסמינר, בדרך כלל מטעם משרד החינוך או מה"ט, במסלולי הנדסת תוכנה, מדעי המחשב ו-QA, בסופן אנחנו משתלבות בעולם ההייטק.

## היציאה לעבודה



כולנו מודעים למורכבות של מציאת משרת ההייטק הראשונה. הפריצה ממעגל הג'וניורים אל תוך החברות מאתגרת בוגרי תואר רבים, ועבורנו החרדיות מהווה מחסום של ממש בהשתלבות בתעשייה.

בשנים האחרונות הושקו מיזמים מגוונים לשילוב של נשים חרדיות בהייטק. חברות השמה ייעודיות לחרדים, קורסים ומפגשי הכנה ופרויקטים להעלאת המודעות. וזה עובד. יותר ויותר חרדיות מתברגות במשרות הייטק בחברות מובילות מתוך הערכה כנה לעבודתן, מטפסות לתפקידים בכירים ותורמות לפיתוח הענף.

## סביבה מותאמת



אולם, מציאת עבודה הולמת היא לא האתגר היחיד העומד בפני בוגרת סמינר. גם לאחר מציאת המשרה המיוחלת קיימים פערים משמעותיים בין שכר עובדת חרדית לשכר עובדת שאינה חרדית הממלאת את אותו תפקיד. לפי דו"ח הכלכלן הראשי של האוצר, ממוצע השכר של העובדות החרדיות נמוך בכ-30% משכר הנשים במגזר הכללי.

יש כמובן את האתגרים הסביבתיים. כידוע, ההלכה עבורנו היא אבן יסוד ועל פיה מושתתים חיינו, מה שלעיתים מייצר קונפליקטים ביחס למקום העבודה, כדוגמת איסור הייחוד עם גברים, ההימנעות מאינטרנט שאיננו מסונן וההקפדה על הצניעות. כל אלה הופכים את השתלבותה של ג'וניורית חרדית בהייטק, למשימה מאתגרת בהרבה מכל בוגר תואר אחר. לשמחתנו, בשלוש השנים האחרונות נכרת עלייה במודעות לשילוב החרדים, וראשי חברות ההייטק המובילות מקדמים גיוס מאסיבי של טאלנטיות חרדיות בוגרות סמינרים, תוך יצירת סביבת עבודה מותאמת עבורן. כך עשרות מפתחות חרדיות סוללות את דרכן אל שוק ההייטק בישראל.

## קצת לאחר שערכנו הכרות, זה הזמן לנפץ כמה מיתוסים



חרדיות עובדות פחות שעות - לא נכון.

מוסר העבודה של החרדיות הוא הגבוהים במשק, בייחוד לאור העובדה שהן פרנסות יחידות. אנחנו נזהרות מאד בגזל זמן ועובדות את מלוא השעות. יש לציין, ללא הפסקות עישון או גלישה באינטרנט.

העובדות החרדיות אינן חברתיות - נו באמת. נשים חרדיות אינן פחות חברתיות מנשים אחרות. ההבדל היחיד הוא שחרדיות

## על הכותבת: מיכל עדיקה-קדושים

הייטקיסטית חרדית.

בודקת תוכנה מוסמכת מטעם ISTQB® עובדת בחברת Yad2.

אוהבת ללמוד ולחקור דברים חדשים תוך כדי תנועה, זה מה שהופך את זה למרגש ומאתגר.

בצד הבודק עם חשיבה של מפתחת ועין של משתמשת. אין יום בעולם התוכנה שלא ניתן ללמוד בו משהו חדש.



מתייחסות למקום העבודה בגישה מקצועית ועניינית בלבד. אנחנו לא מחפשות ליצור קשרים חברתיים, באנו לעבוד.

אז אל דאגה, נח לנו מאד בחברת עצמנו. נשמח שתבינו כשלא נשב אתכם לצהרים.

חרדיות עובדות רק בשביל כסף -

ממש לא! מי שסבור כך טועה בהחלט. זה נכון שאנחנו המפרנסות מבחירה, וחשוב לנו להביא פרנסה הביתה, אבל לא פחות חשובה לנו ההתפתחות המקצועית. אנחנו שואפות להתקדם ולשפר את הידע הביצועים שלנו, ותמיד עם הפנים קדימה ללמוד טכנולוגיות חדשות. אל תשאירו אותנו מאחור בשל מיתוס.

העובדות החרדיות בעלות השכלה נמוכה יותר - לא נכון. אמנם לרוב הנשים החרדיות אין תואר, בשל התנגדות המגזר למסגרות אקדמיות. אך לימודי התעודה בסמינרים, הינם יסודיים ומקצועיים. הם מקנים יכולות לימוד עצמי מפותחות ושמים דגש על פרקטיקה מעשית מעבר לתיאוריה. יש לנו הכשרה מעולה וכלים לעמוד בכל אתגר. נסו אותנו.

זהו להפעם, היה נעים להכיר, ואני מקווה שגיליתם דבר או שניים עלינו, ההייטקיסטיות החרדיות, מזווית אישית.

הולך להיות מעניין.







קובי יונסי

בעל תואר ראשון מאוני' בר אילן בתחום הלוגיסטיקה והכלכלה מוביל את לימודי בדיקות התוכנה במספר מכללות מובילות ביניהן: הטכניון – היח' ללימודי חוץ, מכללת עזריאלי להנדסה בירושלים. מייסד ושותף בהקמת QAMASTERS. מלמד אנשים כיצד לחשוב בדיקות ומסייע לארגונים לשפר מיומנויות של בודקים וללמוד כיצד לחשוב מחוץ לקופסא.



## טבלת החלטות Decision Table

אחת מטכניקות הבדיקה לקופסא שחורה נקראת: טבלת החלטה.

הפעם אתייחס לטכניקה שבוטלת מאוד בחברות פיננסיות, בנקאיות וממשלתיות ואף הופיעה מספר פעמים במבחנים מתוך ראיונות עבודה לבודקים מתחילים. טבלת החלטה היא שיטה לתכנון בדיקות כאשר אנו מתמודדים עם לוגיקה עסקית המורכבת מתנאים שונים שצירופם יוביל לתוצאה מסוימת שאותה נרצה לבדוק. מקורה של השיטה עוד בשנות ה-60 של המאה הקודמת כמענה לצורך של חברות תעופה לפתור בעיות מורכבות של הגדרת מספר רב של צירופים אפשריים בתפעול מטוסים. השיטה מאפשרת לבחור באופן יעיל את מקרי הבדיקה ולהימנע מבדיקות של צירופי קלט לא הגיוניים או כאלו שאינם אפשריים וכך לייעל את תהליך הבדיקות, לזהות מקרים חסרים ולאחר פערי אפיון במפרט הטכני.

### איך השיטה עובדת?

ביסודה של השיטה אנו פורסים בטבלה שורה של תנאים שנקראים גם "תנאים מחוללים" (Triggering Conditions), כל תנאי יוכל לקבל מצב חיובי (True) או שלילי (False), כל שורה תכלול תנאי שיכול או לא יכול להתקיים ובסוף כל עמודה תחכה לנו תוצאה אפשרית.

כל עמודה שיצרנו היא בעצם מקרה בדיקה יחיד שמסכם את אוסף התנאים שיצרו אותו.

### שלבי התהליך בבניית טבלת החלטות

1. לימוד מאפייני המערכת
2. מציאת התנאים והפעולות הנגזרות מהן (תוצאות צפויות)
3. בניית הטבלה על בסיס המשוואה  $N = 2^n$  (מספר מקרי הבדיקה)

לדוגמה במשיכת כסף בכספומט יש לנו 3 תנאים שהמערכת תבדוק:

1. תוקף הכרטיס
2. הקשת סיסמא נכונה
3. יתרה חיובית בחשבון העו"ש

בהתאם לכך יצרתי טבלת החלטה מלאה הכוללת את כל מגוון הקומבינציות:

"כל עמודה שיצרנו היא בעצם מקרה בדיקה יחיד שמסכם את אוסף התנאים שיצרו אותו"

דוגמת הכספומט הבאה, ניתן לצמצם את העמודות הבאות:

עמודות 3-5, 8: ברגע שהכרטיס אינו בתוקף, אין משמעות להתקיימות שני התנאים האחרים (הכרטיס יבלע בלי קשר לסיסמא או ליתרה בחשבון), לכן מספיק לבדוק רק אחד מהצירופים.

עמודות 2, 6: במידה והכרטיס בתוקף, אך הוקשה סיסמא שגויה, הכרטיס יבלע בלי קשר לתנאי היתרה בחשבון. מכאן שמספיק לבדוק רק את אחד משני הצירופים.

צירופי תנאים אפשריים				טבלת החלטה מצומצמת אפשרית:	
7	5	2	1		
T	F	T	T	תנאים	כרטיס בתוקף
T	T	F	T		סיסמא נכונה
F	T	F	T		קיימת יתרה למשיכה
✓			✓	פעולות	הצגת תפריט פעולות
			✓		הפקת שטרות
	✓	✓			בליעת הכרטיס

צירופי תנאים אפשריים								טבלת החלטה מלאה למשיכה מכספומט	
8	7	6	5	4	3	2	1		
F	T	T	F	F	F	T	T	תנאים	כרטיס בתוקף
F	T	F	T	F	T	F	T		סיסמא נכונה
F	F	T	T	T	F	F	T		קיימת יתרה למשיכה
		✓					✓	פעולות	הצגת תפריט פעולות
							✓		הפקת שטרות
✓			✓	✓	✓	✓	✓		בליעת הכרטיס
	✓								הצגת הודעה: "יתרתך הנוכחית אינה מאפשרת משיכה"

### יתרונות השיטה

מומלץ להשתמש בטבלאות החלטה במצבים הבאים בהם השיטה תספק לנו יתרונות בולטים:

1. כאשר אנו מתמודדים עם מערכות מורכבות בעלות לוגיקה עסקית אשר כוללת כמות גדולה של תנאים ותלויות ביניהם (למשל: מערכות בנקאיות, מערכות תומכות החלטה וכד')
2. במקרים בהם אנשי הבדיקות מכירים את הסביבה העסקית ואינם נדרשים לכתיבת מקרי בדיקה מפורטים.
3. במקרים בהם נרצה להציג לצד שלישי (מפתחים, לקוחות, מנהלי פרויקטים, מנתחי מערכות וכד') באופן ברור ופשוט את תוכנית הבדיקות או כיסוי הבדיקות שאנו מכוונים אליו.

הטבלה מתארת את כל המצבים האפשריים של התקיימות התנאים. כל עמודה בטבלה הנה למעשה מקרה בדיקה (test case) המתייחס לצירוף מסוים של תנאים.

מאחר וכל תנאי יכול להתקיים (true) או לא להתקיים (false),

מספר מקרי הבדיקה (צירופים) =  $2 = 2^1$  בחזקת מספר מקרי הבדיקה בדוגמא הנ"ל, ישנם 3 תנאים.  $2^3 = 8$  כלומר סה"כ 8 מקרי בדיקה (צירופים) אפשריים.

כפי שניתן לראות מדובר בטבלת החלטה מלאה המכילה את כל הצירופים האפשריים.

יש בידינו אפשרות גם לצמצם את מספר המצבים ולהתייחס רק למינימום הצירופים הנדרשים לבדיקה. בכדי לעשות זאת, נוריד מהטבלה את הצירופים אשר אינם הגיוניים, יוצרים כפילות, או שאינם



## חסרונות השיטה

1. טבלאות החלטה מספקות כיסוי חלקי ככל שכמות התנאים גדלה באופן משמעותי.
2. מרבית הכלים לתכנון ויישום בדיקות לא תומכים בטכניקה זאת, ויישומה מצריך התאמות או שימוש ביישומים בסיסיים כמו אקסל למימוש התוכנית.

## לסיכום

שימוש בטבלת החלטה היא טכניקה יעילה לתכנון בדיקות במערכות

המכילות לוגיקה מורכבת, לצידן של טכניקות אחרות שסקרנו בטור זה (כמו: ניחוש שגיאות, מקרי שימוש, מעבר בין מצבים וכד')

בשיטה זאת ניתן לפשט תהליכים מורכבים לטבלת מצבים וכך לחשוף את הלוגיקה של המוצר. בפעם הנוספת לאפשר לנו תקשורת טובה יותר עם בעלי עניין נוספים בתהליך הפיתוח.

יישום השיטה דורש רמת מיומנות גבוהה של הבודק והיכרות עם המערכת, זה בא לידי ביטוי בעיקר בתהליך בו יש להמיר את דרישות המערכת לסט של תנאים מחוללים ולקבוע לפיהם את מקרי הבדיקה שנרצה להוציא לפועל.

## בחן את עצמך | טל פאר



### הפתרון לשאלה

נתחיל עם יעד הלימוד ורמת הידע הדרושה לנושא הזה. את יעדי הלימוד ניתן למצוא בעמוד הראשון בכל פרק. לכל יעד לימוד מוגדרת רמת הידע הדרושה לו (K-level).

יעד הלימוד שלו מתאימה שאלה זו היא FL-1.3.1 "הסבר את שבעת עקרונות הבדיקה" והיא ברמה K2. רמה זו אומרת שעל הנבחן להבין את התשובה הנכונה על פי הנושא הנלמד.

כאמור, עקרונות הבדיקה המוצגים בסילבוס הוגדרו על ידי מומחי בדיקות לאורך שנים, בהתבסס על ניסיון רב בתכנון וביצוע בדיקות. עקרונות אלה מראים, למשל, שככל שנקדים את תחילת הבדיקות (early testing) נגלה פגמים בשלב מוקדם יותר בתהליך הפיתוח (למשל, זיהוי פגמים בדרישות) וכך נחסוך זמן וכסף מאחר והפגמים יתוקנו עוד לפני כתיבת הקוד ולפני שהמערכת נמסרת לבדיקות.

עיקרון אחר מדגיש שגם אם ביצענו בדיקות כראוי ואנחנו (יחד עם צוות הפרויקט) מרגישים שהמערכת מוכנה למסירה, הרי שיתכן והמערכת לא באמת עונה על דרישות הלקוח.



### הבה נבחן את התשובות:

עיקרון 5 (פרדוקס ההדברה, pesticide paradox) מזהיר מחזרה על אותן בדיקות שוב ושוב. אם נבצע את אותן בדיקות שוב ושוב הן לא יגלו פגמים חדשים מאחר הבדיקה מצאה את כל הפגמים שהיא יכולה למצוא. כדי להימנע מכך וכדי למצוא פגמים חדשים יש לשנות את הבדיקה (לדוגמה, להשתמש בנתונים אחרים או לשנות את סדר הצעדים בבדיקה) או לכתוב בדיקות חדשות. זוהי אינה תשובה נכונה.



שילוב הבודקים בשלב מוקדם בתהליך הפיתוח הוא צעד נכון, שכמו שהצגתי למעלה, עשוי לזהות פגמים בשלב מוקדם ולהביא לחיסכון בזמן וכסף. אולם זה לא אומר שבכך לא יוצגו פגמים במערכת או שזה יסייע במציאת כל הפגמים. זוהי אינה תשובה נכונה.



סביר להניח שאם יתאפשר לצוות הבדיקות לבצע בדיקות ממצות (exhaustive testing) יימצאו במערכת פגמים נוספים שלא נמצאו עד כה, אולם זה בלתי אפשרי לבצע בדיקות ממצות ולכן תשובה זו אינה נכונה.



כל זמן שצוות הבדיקות מבצע בדיקות הרי שבסבירות גבוהה (עד מאוד) יימצאו פגמים חדשים. מציאת פגמים מוכיחה שקיימים פגמים במערכת, אך גם מציאת פגמים רבים אינה יכולה להוכיח שבסיום התהליך אין פגמים נוספים במערכת. זהו העיקרון הראשון ברשימת העקרונות שבסילבוס "הבדיקות מצביעות על נוכחות של פגמים, לא על היעדרם" (presence of defects).



לפי ההסברים הנ"ל התשובה הנכונה היא תשובה ד'.

אם תרצו שאגיש שאלת דוגמה בנושא מסוים או אם יש לכם שאלות אנא פנו אלי באימייל [tal.peer@practical-testing.com](mailto:tal.peer@practical-testing.com)



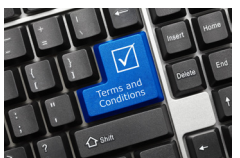
רקע

מסמך אפיון הינו, תוצר סובייקטיבי של ניתוח המערכת - אחד השלבים בתהליך הקמת מערכת מידע. מנתח המערכות מבצע פעולות ניתוח וחקר לגבי מערכת המידע הנדרשת (או הרכיב הנדרש) מול הלקוח והתוצר הינו 'מסמך אפיון' המתאר את הרצונות והצרכים של הלקוח. המאפיין, מנהל הפרויקט, צוותי פיתוח והבדיקות, צוות אינטגרציה, צוות נתונים וכיו"ב (להלן "בעלי העניין") פועלים כל אחד בגזרתו בהתאם למסמך האפיון מתוך הנחת עבודה שהמסמך אכן מתאר, עד כמה שניתן, את דרישות הלקוח בדיוק מירבי. המאפיין צריך לוודא שכל בעלי העניין אשר קראו את מסמך האפיון, הבינו אותו באופן דומה, כאשר כל בעל עניין יודע לקחת מהאפיון את המרכיבים הרלוונטיים לצורך פיתוח המערכת ו/או בדיקתה.

את גישת הבדיקות לניתוח של מסמך האפיון ניתן לחלק ל-4 שלבים עיקריים\*:  
שלב ראשון: קבלת האפיון - הבודק פוגש לראשונה את האפיון ומבצע מעבר ראשוני על האפיון. שלב שני: חלוקת האפיון - הבודק מחלקת את האפיון לנושאים לבדיקה ונושאים לא לבדיקה. שלב שלישי: ניתוח הנושאים לבדיקה - כולל מרכיבים אשר לא נמצאים באפיון הדורשים בדיקה. שלב רביעי: בקרת תוצרים.

\*ישנם שלבים נוספים ותתי שלבים מבחינת אופן ניתוח אפיון מערכת, מכיוון שיריעת הכתבה קצרה מלהכיל את כל המידע, בכתבה זו אתייחס לשלבים אשר לטעמי מהווים חלק מאבני הדרך הנדרשות לכל בודק תוכנה בבואו לנתח אפיון מערכת.

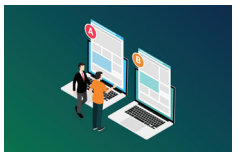
שלב ראשון - קבלת מסמך האפיון



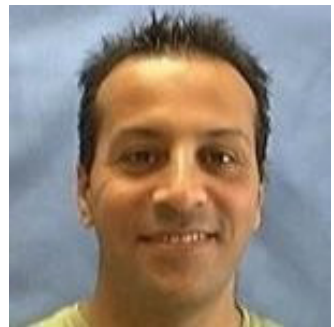
הבודק פוגש לראשונה את מסמך האפיון לאחר שמנתח המערכות סיים לבצע את כתיבת האפיון והאפיון מוכן לבקרה של בעלי העניין במסגרת CDR או DR. הבודק נדרש במסגרת פגישות ה-Review על האפיון להגיע ברמת מוכנות גבוהה בעיקר בשלב ה-DR, כאשר הוא קרא את האפיון, הבין את התהליכים העסקיים, יצר רשימת שאלות לגבי היבטים לא מובנים או לא ברורים באפיון, למשל חסר מידע מסוים בכדי להבין רכיב במערכת או שיש חוסר הגיון בתהליך עסקי מסוים. את רשימת השאלות הבודק שולח מראש למנתח המערכות או שמעלה במהלך פגישות ה-Review וכך מצופה שיעשה ע"י כל בעלי העניין.

מנתח המערכות, לאחר קבלת כל ההערות מעדכן את האפיון בהתאם ולאחר מנגנון של אישורי אפיון הוא מעביר את המסמך לתחילת עבודה של בעלי העניין. הבודק (אשר מכיר את האפיון במסגרת ה-CDR/DR) מגיע עם הבנה גבוהה לגבי מבנה המערכת, הפן העסקי, מחוזק בתשובות אשר קיבל לשאלותיו ויכול לגלוש לשלב השני ביתר קלות, כאשר רכיבי המערכת ומהות המערכת מובנת ובהירה יותר.

שלב שני - חלוקת האפיון



האפיון אשר עבר את שלבי האישור הדרושים (לקוח, מנהל פרויקט) מגיע ברמת בשלות גבוהה לבודק, כאשר על הבודק לבצע מעבר נוסף על האפיון ולחלק אותו לשניים או שלושה חלקים: נושאים שיבדקו ונושאים שלא יבדקו (לפעמים ניתן להוסיף חלוקה נוספת "נושאים בצריך עיון").



ליאור שם טוב

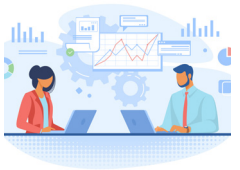
ניהול פרויקטים וייעוץ בתחום הבטחת איכות מערכות מידע מעל ל-20 שנה.



מדוע חלוקת האפיון נדרשת? החלוקה מייצרת סדר באפיון באמצעות מיפוי הדרישות הרלוונטיות לבדיקות, כך שבמידה והבודק, יימצא שבמיפוי חסרות דרישות או שהדרישות חלקיות הוא יוכל להביא לידיעת מנתח המערכות את הממצאים. מציאת הפערים בשלב זה הוא קריטי מבחינת פיתוח המערכת, מהסיבה שמנתח המערכות יכול לבצע בשלב מוקדם שינויים נדרשים ולעצור את הפיתוח בזמן. ייתכן שישנם היבטים במערכת אשר משפיעים בתוך המערכת ו/או מחוצה למערכת, במצב זה יש לחזק את הרשימה של הנושאים שיבדקו ולציין איזה רכיב משפיע או מושפע ולשלב בבדיקות את האלמנטים המשפיעים על מערכות אחרות או מושפעים ממערכות אחרות.

חשיבות לא מבוטלת יש לתת לנושאים שלא ייבדקו. רשימה זו מציגה תמונת מצב לבעלי העניין אילו דרישות או נושאים ו/או תתי נושאים באפיון הוחלט ע"י הבודק לא לבדוק, בנוסף לכך על הבודק לציין את הסיבה שאותו עניין לא ייבדק. בעלי העניין נדרשים להגיב למסמך זה ובהתאם לכך מתקבלת החלטה אם יש נושאים אשר יעברו לצד הנושאים שיבדקו או לא (ייתכן שגם בצד הנושאים שיבדקו יהיו תתי נושאים אשר שאין צורך בבדיקתם)

שלב שלישי - ניתוח נושאים הנדרשים לבדיקה



לאחר שהבודק סיים לגזור ולמפות את בדיקות המערכת הנדרשות יחל שלב הניתוח של כל דרישה ודרישה שבה יש להבין מהדרישה את המידע אשר מרכיב את הפתרון העסקי, למשל, חוקים עסקיים, תיאור פונקציונאלי של תהליך עסקי, טבלאות חדשות, הוספת שדות לטבלאות קיימות, תצוגת מסך, תעבורת קבצים, מידור ועוד. כמו כן בודקים אם ישנם רכיבים בדרישה אשר עומדים בפני עצמם או משפיעים על רכיבים אחרים בדרישות אחרות או מושפעים מהם.

הבנת הדרישה ופירוקה מוביל את הבודק לסווג אילו סוגי בדיקות מתאימים להיבטים השונים המרכיבים את הדרישה, למשל: בדיקות פונקציונאליות, בדיקות מסדי נתונים, ממשק משתמש, בדיקות ולידציה, הרשאות ועוד. אמנם כל דרישה מקבלת טיפול פרטני, אבל יש לתת את הדעת על דרישות עם קשרים ביניהם ולטפל בהן באותו אופן של ניתוח האפיון מבחינת היבטים המרכיבים את הקשר וסיווג הבדיקות.

סוגי הבדיקות מהווים לבודק תמונה מקדימה והערכות מראש, טרם הגעת המערכת לסביבת הבדיקות. הבודק יכול לתכנן את עץ הבדיקות, מקרי הבדיקה ותסריטי הבדיקות המושפעים מסוגי הבדיקות אשר נגזרים מדרישות המערכת וגם להכין נתוני הבדיקה לתסריטי הבדיקות.





**סיכום**

קריאת מסמך אפיון צריכה להתבצע בצורה מסודרת על ידי הבודק, כהערכות ל-CDR לבצע קריאה ראשונית כדי לקבל רקע כללי והבנת הצורך של הלקוח. בשלב הקריאה הראשונית הירידה לפרטים אינה חשובה, אלא מטרתה לנודד לאורך האפיון ולפגוש את מרכיבי המערכת, התהליכים העסקיים, ממשקים למערכות הפנימיות והחיצוניות ועוד.

לאחר קבלת הרקע הכללי, אנו מוכנים לשלב הבא בניית האפיון. על הבודק לעבור על האפיון על כל חלקיו, ורכיביו ולהבין כל חלק על בוריו לקראת פגישת ה-DR. אלמנטים שלא מובנים יש לצייןם. בודק אשר מתעלם מנושאים, משפטים או מילים לא מובנות ולא מסמן אותם כשאלות למאפיין, מסכן את פרויקט הבדיקות. הכלל אומר: 'מה שמובן באפיון יבדק ומה שלא מובן "כנראה" שלא יבדק'. נושאים לא מובנים מהווים חוליה חלשה בתוך האיכות ויש לטפלה באמצעות פנייה לאחד מבעלי העניין אשר ייתן ביאור לאותו נושא וכך הבודק יוכל לבחון אם אותו נושא נדרש לבדיקה או לא.

לאחר מיפוי הדרישות הבנת הצורך העסקי וחלוקת הדרישות למה יבדק ומה לא יבדק, הבודק מוודא שכל ההיבטים העסקיים מנגנים נכון בין הדרישות ובתוך חלקי האפיון. כל הנושאים הלא מובנים מקבלים ביאור וכל תוצרי ניתוח האפיון עוברים לבקרה של בעלי העניין להערות. במהלך חיי הפרויקט הבודק צריך להיות ער לשינויים באפיון ולבצע את ההתאמות הנדרשות כדי לשמור על רמה גבוהה של איכות המידע אשר הוא מסתמך עליו עד לסיים הפרויקט.

**שלב רביעי - בקרה**



בודק תוכנה הוא גוף ביקורתי אשר מטרתו לבצע בקרה על איכות המערכת ולתת את חוות דעתו המקצועית, אך על הבודק להיות מבוקר גם כן, כלומר לשקף לבעלי העניין את תוצרי ניתוח האפיון כדי לקבל מהם הערות. מטרת ההערות של בעלי העניין היא לתת לבודק נקודת מבט נוספת, כי ייתכן שהוא פספס דרישה במיפוי ו/או חוסר בטיפול בסוגי בדיקה נוספים או ייתכן שיש גם חוסר הבנה מבחינה עסקית או בדרישה מסוימת או יותר.

הבקרה שומרת על הבודק, היא גם יוצרת אחריות משותפת ביחד עם כל בעלי העניין אשר חשופים לתוצריו. ההערות אשר התקבלו מבעלי העניין, עוברות לבודק על מנת לבצע את ההתאמות הנדרשות בתוצריו וכך הוא יוצר מעין תמונה שלמה של תוצרי הדרישות.

**'מה שמובן באפיון יבדק ומה שלא מובן "כנראה" שלא יבדק'**



צוות המעצין מחפש אתכם...  
רוצים לקחת חלק במעצין?  
אנו מחפשים אחראיות לסור  
"ראיון עם מנהלות בדיקות"

למעוניינים יש  
ליצור קשר עם ניצן:

INFO.TESTINGWORLD@GMAIL.COM  
בצירוף פרופיל הלינקדאין שלכם



תמרה מוסונובה

בודקת תוכנה ואינטגרציה בחברת Varonis. בעלת תואר בתקשורת וקולנוע והסמכות פיתוח אפליקציות Web של מיקרוסופט, כך משלבת חשיבה יצירתית ואנליסטית בעבודה. בונה אתרים ועורכת סרטים כתחביב. שחקנית כדורשת בליגה ארצית, תופסת כדורים ובאגים מקצועית.



אסתר צבר

מהנדסת (M.Sc.) בעלת 23 שנות ניסיון בפיתוח ובדיקות תוכנה, מתוכן 11 שנים בניהול QA בחברות BMC | ECI ובנוסף חברה ב-Advisory Board של ITCB הארגון הישראלי להסמכת בודקי תוכנה. בתשע השנים האחרונות – יזמית ומנהלת של AQA המכשירה ומשלבת אנשים עם תסמונת אספרגר בעבודה בהייטק כבודקי תוכנה.



שי ביטון

בעל 12 שנות ניסיון בפיתוח אוטומציות ובדיקות. עובד כיום ב-Qwilt.



טל פאר

בעל ניסיון של יותר מ-20 שנים כבודק ומנהל בדיקות במגוון חברות וטכנולוגיות במודלי פיתוח שונים. כיום טל יועץ ומדריך בדיקות עצמאי.

טל חבר ב-ITCB וגזבר הארגון העולמי ISTQB.



עמית ורטהיימר

בודק תוכנה ב-Deep Instinct.



רוביק סביאנץ

בודק תוכנה, נמצא בתחום מעל 4 שנים את דרכו התחיל בחברת CARAMBOLA נכון להיום מחזיק את מערך הבדיקות בחברת OOLO בזמן הפנוי - ספורט, טיולים ומחשבים



אפרת וינברג

עוסקת בבדיקות תוכנה קרוב ל-20 שנה. עבדה במספר ארגונים בתפקידי בדיקות וניהול בדיקות. בשנים האחרונות עוסקת בפיתוח והוראת קורסים בבדיקות תוכנה ונושאים נוספים.



משה מאמיה

בעל 17 שנות ניסיון כמהנדס, מתוכן מעל עשר שנות ניסיון ניהול, מתמחה בפיתוח אוטומציה ובבדיקות ביצועים. עובד מעל 5 שנים ב-HP כמנהל קבוצות QA ו-DevOps. חבר מייעץ למועצת מנהלים של ISTQB ומרצה בפקולטה להנדסת תוכנה במכללת SCE.



רחל ברוך

הנדסאית תוכנה, לפני 3 שנים לאחר הפסקה לעולם התוכנה ההייטק חזרה לעולם התוכנה בכל הכוח. נהנית להיות בצד הבודק עם חשיבה של מפתחת. אין יום שהיא לא לומדת משהו חדש בעולם התוכנה.



שירה נוסבויים

הייטקיסטית ואמא במשרה מלאה, בדקות הבודדות שנשארות ביום בלוגרית אפייה. בעלת תואר ראשון במדעי המחשב ובכימיה, מעל 10 שנות ניסיון כמפתחת תשתיות אוטומציה וכלים אוטומטיים בחברות גדולות, בסטארטאפים שונים בתעשייה במגוון תחומים. מובילת תחום, מרצה ומפתחת קורסי אוטומציה.

בשבילה החיים זה לא מספיק.



יאיר נסימוב

בעל ניסיון של כ-10 שנים בתחום בדיקות אוטומטיות. מייסד חברת Rain the Dog ומתן פתרונות אוטומציה מותאמים אישית לחברות. מרצה ומפתח קורסי אוטומציה. יוטיובר מתחיל אספן תקליטים, חובב טיולים, ריצה ואגרוף תאילנדי נשוי +2 + כלב - Rain.



## למה לכם לחשוב אם פספסתם?!?

הרשמה

אם אתם רוצים שהגיליון הבא של מגזין עולם הבדיקות יגיע אליכם - לחצו על הלינק להרשמה

[bit.ly/TW-Reg](http://bit.ly/TW-Reg)