

רבעון רביעי 2023 גיליון מס' 35

הבדיקות



עולם ^{מגזין}

www.testingworld.co.il

בדיקות מבוססות מודל
ד"ר מיכאל בר סיני

בואו נדבר על הפסיכולוגיה
של הבדיקות
משי מוטולה

האנציקלופדיה לבדיקות
קובי יונסי

בדיקות ריגרסיה – גישות,
מצבים ופתרונות
ניסים אריאל

דבר העורך ניצן גולדנברג

מגזין
עולם
הבדיקות



ניצן גולדנברג

מזה 8 שנים בתחום
בדיקות התוכנה, תפקיד
נוכחי מהנדס בדיקות
בכיר בחברת **SeatGeek**,
מנהל קבוצת ה-AB של
ארגון **ITCB®**
המוביל הראשי של קבוצת
המיטאפ **TestIL**, מרצה
בקורסים לבודקי תוכנה



קוראים יקרים,
מונח לפניכם גליון מס 35 של מגזין "עולם הבדיקות".
בגליון זה תוכלו למצוא מגוון רחב של מאמרים חדשים, בנוסף לטורים המעולים והקבועים שלנו:
"בדיקות מבוססות מודל" - ד"ר מיכאל בר סיני
ניסים אריאל מביא לנו עוד מאמר מעניין בנושא "בדיקות ריגרסיה - גישות, מצבים ופתרונות"
משי מוטולה תספר לנו על פסיכולוגיה בבדיקות במאמר "בואו נדבר על הפסיכולוגיה של הבדיקות"
יש לנו תשבץ חדש ומאתגר בשבילכם בגליון הזה ואנו עוד מחכים לקורא שישלח לנו את הפתרון של
התשבץ מגליון 34

הקשבנו לבקשות שלכם ואנו מחזירים את הטור שאהבתם "בחן את עצמך"
בנוסף, יש לנו את הטורים הקבועים: מחפש צרות, ראיון עם מנהל/ת בדיקות, האנציקלופדיה לבדיקות, עושים
QA לקריירה, משולחנו של שביט ונגיסה מ-TestIL ובודקים בכיף.

אנו נשמח לקבל בקשות לנושאים חדשים ומעניינים למאמרים, צרו עימנו קשר במייל

magazine@testingworld.co.il

קריאה מהנה,
ניצן גולדנברג

you can do
everything

עדכון מוסמכים בישראל

סה"כ מוסמכים בבורד הישראלי **ITCB®** - 9,903
מתוכם 852 "מצטיינים" אשר ענו מעל 90% תשובות נכונות.
106 מצטיינים ברבעון השלישי של שנת 2023

**ITCB® ומגזין
"עולם הבדיקות"**
מרכינים ראש
לזכר הנופלים
ומבקשים לחזק את
משפחותיהם של
הנופלים, הנעדרים,
הפצועים והחטופים
ואת אלו שפועלים
למען ביטחון והגנת
המדינה.



תוכן העניינים

- 2.....דבר העורך
- 4.....בחן את עצמך - שאלה | **ירון צוברי**
- 5.....משולחנו של שביט - עדיף באג אחד מתועד מאשר 2 באגים באוויר | **שביט ג'רסי**
- 6.....בדיקות מבוססות מודל | **ד"ר מיכאל בר סיני**
- 10.....ראיון עם מנהלת בדיקות - פריון | **קארין בוחבוט**
- 11.....האנציקלופדיה לבדיקות - "מדדי בדיקות תוכנה" | **קובי יונסי**
- 13.....בדיקות רגרסיה - גישות, מצבים ופתרונות | **ניסים אריאל**
- 15.....עושים QA לקריירה - "לצאת מאזור המצויינות" | **איילת מלמד**
- 16.....בחן את עצמך - תשובה | **ירון צוברי**
- 17.....מחפש צרות - "תרים טלפון" | **מיכאל שטאל**
- 19.....בואו נדבר על הפסיכולוגיה של הבדיקות | **משי מוטולה**
- 21.....נגיסה מ-TestIL
- 22.....בודקים בכיף
- 23.....דף העורכים

מו"ל
 Israeli Testing Certification Board
 ITCB®

ניהול המגזין
 iMDsoft, ברון, יאן

ניהול התוכן
 קובי הלפרין, Red Hat

עורך ראשי
 ניצן גולדנברג, SeatGeek

עיצוב גרפי
 בית נלי מדיה
 סטניסלב קולנקו
www.beitnelly.com

יצירת קשר
 אימייל:
magazine@testingworld.co.il

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



www.testingworld.co.il



www.itcb.org.il

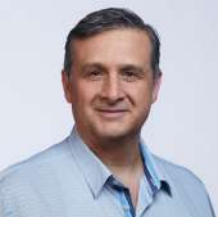


**עולם הבדיקות נכתב ע"י בודקים
 עבור בודקים**

**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 מאמרים. 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 magazine@testingworld.co.il.



ירון צוברי

מעל 30 שנות ניסיון בפיתוח תוכנה, ניהול פרויקטים מורכבים, הנדסת מערכות ובדיקות. עם ניסיון בינלאומי במערכות מורכבות בתחומים: טלקום, פיננסים, אוטומוטיב ומערכות הגנה. מומחיות עיקרית: ניהול פרויקטים מורכבים, ייעוץ להנהלה בכירה בארגונים (Siemens Germany), אימון מנהלים. נשיא וסגן נשיא ארגון ISTQB® לשעבר ונשיא ITCB® מיום הקמתה.



אחת המשימות החשובות של בודקי התוכנה היא לבצע ניתוח (Analysis - אנליזה) לכל סוגים שונים של נתונים או מקרים ובין השאר לבצע ניתוח וסקירה (review) לדרישות ו/או למידע על הדרישות - בעיקרון לכל דרישה שהיא, בין אם היא מגיעה מהמכירות, מהלקוחות, ממנהלי המוצר, מהארכיטקטים של המערכת, המפתחים או כל אחד שאחראי לייצר ולהגדיר אותן.

על מנת להיות יעילים בביצוע הניתוח והסקירה לדרישות, רצוי לנצל מידע וניסיון קודמים שנצברו: מסקירת פרויקטים קודמים, ממופעים קודמים של הפרויקט, ממוצרים דומים או שונים, ממידע הקיים בספרות המקצועית, מהאינטרנט, ממידע שנרכש בהכשרות מקצועיות (קורסי בדיקות תוכנה למשל), ולאחרונה ממידע שכלים כמו ChatGPT או Bard מספקים לנו (במידה ונדע כמוכן בין היתר איך לבקש זאת מהם), וכו'. ניסיון שעיקרו: על מה והיכן לשים דגש בזמן ביצוע הניתוח ו/או הסקירה? לדוגמא: מידע חסר, מידע מנוגד בין דרישות, דרישה לא הגיונית, דרישה שאינה ברת בדיקה (not testable); מאפייני איכות שחסרים או לוקים בחסר בדרישות (לדוגמא: רשימת דרישות לביצועים שלהם המערכת נדרשת) וכו'. אחת מהדרכים לצבור ולשמר ידע זה באופן יעיל הוא לאגד את כל ה"ל" בתוך רשימות תיוג (checklists) שימשו בזמן ביצוע האנליזה ו/או הסקירה לדרישות.

ועכשיו לשאלה:

איזה מהמשפטים הבאים המיוחסים לשימוש ברשימות תיוג (checklist) בסקירה רשמית (formal review) רשמית הוא נכון?

- א. כחלק מתכנון הסקירה, הסוקרים יוצרים את רשימות התיוג הדרושות לביצוע הסקירה
 - ב. כחלק מהתקשורת בנוגע לסקירה, הסוקרים ממלאים את רשימות התיוג שסופקו עבורה
 - ג. במסגרת ישיבת הסקירה הרשמית יוצרים הסוקרים דוחות ליקויים על סמך רשימות התיוג שסופקו עבור הסקירה
 - ד. במסגרת ייזום הסקירה מקבלים הסוקרים את רשימות התיוג הדרושות לביצוע הסקירה
- בחר אפשרות אחת.

*לצפייה בתשובה המפורטת - דפדפו לעמוד 16

הצטרפו לצוות המוביל את מגזין עולם הבדיקות

מעוניינים להצטרף לעשייה? התפנה מקום בצוות המגזין!

הפעילים במגזין הינם אנשי מקצוע בתחום הבדיקות שפועלים בהתנדבות למען קהילת הבודקים בארץ.

לקבלת פרטים נוספים פנו לניצן גולדנברג: magazine@testingworld.co.il



שביט ג'רסי

אבא לתאומים בני 3, מנהל בדיקות בחברת Wisetamp, בעל תואר ראשון בהנדסת תעשייה וניהול מהטכניון, בעל 11 שנות ניסיון בתחום הבדיקות, מתוכם מעל 8 שנים מדריך עצמאי ל-QA. בעל סדרת הסרטונים השבועית "QA ללא הפסקה"



בכתבה הזו אני רוצה לדבר אתכם על מקרים שגרתיים שקורים לנו ולא מעט כבודקים.

זה משהו שאני שם לב שקרה אצלי בעבר הרחוק כבודק צעיר ומתחיל וגם היום כמנהל ובעבודה אל מול צוותים שונים.

זה קורה בדרך כלל כשאנחנו מתחילים לעבוד עם מוצר לא מספיק יציב או פיצר חדש ששוחרר זה עתה לבדיקה או אפילו פיצר שעדיין לא שוחרר לבדיקה אבל המפתח העלה אותו לסביבת בדיקות מתוך הסביבה הלוקאלית שלו כדי לבצע מס' פעולות שונות ואנחנו מתחילים לעשות מעט מחקר מקדים. כדי ללמוד את המוצר מעבר לקריאה, ניתוח ומעבר על האפיון הכתוב.

כלומר, אנחנו רוצים מעט לשחק או להרגיש את המוצר לפני שאנחנו מתחילים לכתוב עליו טסטים במסגרת האפיון.

מה שקורה בדרך כלל הוא שאנחנו מתחילים להריץ בדיקות ולעשות פעולות שונות במוצר שלנו ותוך כדי אנחנו רואים משהו לא עובד כשורה. אחר כך תוך כדי אנחנו תופסים עוד משהו ואחריו עוד משהו... ומהר מאוד אנחנו מקבלים רושם לא טוב. מקבלים את הרושם שיש כאן מוצר לא יציב. לא אפוי.

החטא הגדול של הרבה בודקים הוא שאינם עוצרים לרגע ושואלים שאלות, לא מנסים להבין האם באמת מדובר בכמה בעיות שונות, מהי חומרתם?, ומהו שורש הבעיה?, האם בעיה מסוימת גוררת בעיות אחרות?, האם כל הבעיות נוצרות בגלל בעיה אחת? ועוד ועוד כהנה שאלות.

זה עוד בכלל לפני שהתייחסנו לאפיון, לפני שהלכנו להתייעץ עם המפתח שפיתח את זה.

יש כאן כמה בעיות בהתנהלות השוטפת שלנו כבודקי תוכנה. אבל אחת הבעיות שמטרידות אותי יותר מכל היא שאנחנו לא עוצרים כאן. מתעדים לפחות בדרך אחת את הבעיות שחווינו ואחר כך ממשיכים הלאה, לשאול, לתחקר, להתייעץ וכו'.

מדוע זה מטריד אותי? זה מטריד כי זאת הדרך לאבד את התקלות שמצאנו ולהתקדם בכלל עם בדיקות אחרות לגמרי במיוחד כשמדובר בתקלות שהן לא ברורות מאליו ואולי גם מקרי קצה. לפעמים אנחנו הולכים צעד אחד רחוק מדי עם התחקור על אותה תקלה. מגלים עוד כמה הפתעות בדרך אבל למעשה לא תיעדנו דבר ממה שראינו במקרה טוב חצי שעה-שעה קודם לכן.

ואני לא מדבר על ממש לתעד את הבאג באופן רשמי במערכת לניהול הבאגים..

אני מדבר על שימוש בכלי שיתעד בדרך זו או אחרת את התקלה ויזכיר לנו שהתחלנו את התחקור מתקלה מסוימת.

זה יכול לקרות בצילום מסך, צילום וידאו או אפילו כלי שיושב על הדפדפן ומקליט את הדברים שאנחנו עושים וחווים באותה יממה או מהשעות האחרונות. גם זו דרך לשמור בצד את התקלות שחוונו כדי שנוכל אחר כך בשלב מאוחר יותר לחזור ולתעד את התקלה באופן רשמי.

עם זאת, אנחנו צריכים לזכור שאנחנו צריכים לבצע את המוטל עלינו, לעמוד בזמנים ובכל זאת...אף אחד מאיתנו הוא כנראה לא עובד במשרה זמנית כ-להטוטן שמנסה לתפוס את כל הכדורים באוויר ביחד ולשלוט בכל.

לכן אנחנו מחויבים לעבודה במתודה סדורה וב-צ'אנקים קטנים. תבדקו חלק-חלק מתוך הפיצר המלא וכשיש בעיה או אפילו מס' בעיות. שם תעצרו.

תוודאו שזה מתועד איכשהו באופן לא רשמי... תתקדמו הלאה. וכן חברים.. גם אם זה מעט קוטע לכם איזו המשכיות או קו מחשבה של כיוון מסוים..

זכרו – אתם עדיין ביתרון גם אם הלכתם 2 צעדים קדימה וצעד אחורה.

אני מאחל לכם המשך שבוע נעים ושקט, שנדע ימים טובים יותר ושלעולם לא תצעדו לבד.



במאמר זה, אנו נתמקד במושג MBT= Model Based Testing, או בעברית - בדיקות מבוססות מודל. כיצד מידול של מערכות שאנו בודקים משדרג את תפקיד הבודק?

מהו מודל?

מודל הוא ייצוג מופשט ושימושי של מערכת מורכבת. "מופשט", כי הוא כולל רק את הפרטים המעניינים במערכת המתוארת. ו-"שימושי", כי ניתן לבצע עליו פעולות בעזרת מחשב. בעולם הבדיקות, מודלים מסייעים בהבנת התנהגות המערכת הנדרשת, ואת הקשרים והאינטראקציות בין רכיביה השונים. תיאור זה צריך להיות פשוט מספיק בשביל שניתן יהיה לעבוד איתו ולהבין אותו, ועם זאת מפורט מספיק על מנת שהתוצרים המופקים ממנו יהיו שימושיים.

לדוגמה, ניתן לבנות מודל המתאר איך המערכת אמורה להתנהג כאשר משתמשים מבצעים בה פעולות מסוימות. ממודל כזה ניתן להפיק ייצוג ויזואלי פשוט של ההתנהגות הרצויה של מערכת תחת בדיקה, וכך לעזור לבודקים להבין, לנתח ולחזות את התנהגותה של המערכת. בנוסף, אפשר לייצר ממנו תוכנית בדיקות, ואפילו לבדוק אם יש התנגשות בין הדרישות השונות.

כדי להמחיש טוב יותר את הקונספט של מודלים בבדיקות תוכנה, נתייחס לדוגמה המוכרת לכולנו מחיי היום יום - התמצאות בעיר. דמיינו שמישהו עוצר אתכם ברחוב ושואל איך מגיעים לתחנת הרכבת הקרובה. ישנן שתי גישות אפשריות להכווין אותו לשם:

גישה ראשונה - אתם מתארים לו מהם הצעדים שעליו לבצע על מנת להגיע לשם (המשך ישר במשך 200 מ', פנה שמאלה בכיכר, תמשיך עוד 500 מ', פנה ימינה וכו'..).

גישה זו מייצגת **גישה תכנותית**, היא מתמקדת בריצה אחת, ברשימת הוראות ספציפית. גישה זו משקפת את שיטת הבדיקות הרווחת היום - בדיקות מבוססות תסריטים (scenario-based).



גישה שנייה - אתם מראים לו מפה של רחובות העיר ומסמנים לו את מיקומו הנוכחי ואת מיקום תחנת הרכבת. המפה מייצגת **מודל המספק מבט כולל על רחובות העיר**, הצמתים ונקודות הציון השונות. היא מאפשרת לכל אחד לראות את כל המסלולים האפשריים, לתכנן את המסלול שלו, לחקור מסלולים אלטרנטיביים (ואולי אפילו תחנות רכבת אלטרנטיביות) ולהבין טוב יותר את מבנה העיר הכולל.



גישה זו משקפת את שיטת בדיקות מבוססות מודל (model-based testing ובראשי תיבות MBT).

שיטת MBT מתבססת על בניית מודל בדיקות - מעין "מפה" של כל הבדיקות האפשריות של המערכת. באמצעות בחינת מסלולים שונים בשילובים שונים, נוכל להבין את המערכת בצורה רחבה וטובה יותר ולבדוק יותר תסריטים, כולל כאלו שאולי לא היינו מודעים אליהם אפילו ללא המודל. אבל, לא תמיד יש לנו מפה שנוכל להעביר, וגם כשיש, כנראה שיש גם הרבה מסלולים לבחור. ואולי בכלל רחובות העיר עברו בנייה אינטנסיבית לאחרונה והמפה כבר אינה מעודכנת?



ד"ר מיכאל בר סיני

מיכאל מתמחה בהנדסת תוכנה, בדגש על ארכיטקטורת מערכות, שפות תכנות, ומפרטים פורמליים בעל ניסיון של למעלה מ-25 שנה בפרויקטי פיתוח תוכניות מגוונים בארץ ובחול"ל. לפני Provento, מיכאל עבד במכון למדעי החברה הכמותיים של אוני' הארווארד (IQSS) כחוקר אורח. בין השנים 2012-2020 מיכאל היה הבעלים של חברת ייעוץ לפרויקטי תוכנה, ועבד באוניברסיטת בן-גוריון בנגב כדוקטורט וחוקר בתחום הנדסת התוכנה.



"שיטת MBT מתבססת על בניית מודל בדיקות - מעין "מפה" של כל הבדיקות האפשריות של המערכת"

כלי המידול הוא מעיין אפליקציית ניווט בין כל הבדיקות האפשריות של המערכת (כן, יכולות להיות הרבה כאלו). ככל שאנו מתקדמים בבנייה והבנה של המערכת הנבדקת, אנו מוסיפים למודל הנחיות ודרישות. המודל, מצידו, מעדכן את "מפת הבדיקות" האפשריות, ויועד לייצר לנו באופן אוטומטי תסריטי בדיקה חדשים, המייצגים את המסלולים החדשים שנוצרו והשילובים ביניהם.

בעולם ה-QA אנחנו בדרך כלל לא מעוניינים בבדיקה אחת, אלא בחבילת בדיקות המכסה חלקים במערכת הנבדקת. אם נמתח את הדוגמה קצת, זה יהיה כמו לבקש מאפליקציית הניווט שלנו לייצר "חבילת מסלולים, העוברים בתחנת רכבת, תחנת אוטובוס, ועמדות טעינה לרכב חשמלי, כאשר חלק מהמסלולים משתמשים בכבישי אגרה וחלק לא". אבל זה באמת כבר קצת למתוח את הדוגמה יותר מדי. בואו נחזור לעולם הבדיקות.

איך מודלי בדיקת תוכנה משפרים את התקשורת בארגון?

מודלים משמשים **כשפה משותפת** בין המחלקות השונות המעורבות בתהליך פיתוח התוכנה. זאת משום שהם מאפשרים ייצוג ויזואלי של התהליכים וההתנהגויות השונות של המערכת. בגדול, מודלים לוקחים דרישות מערכת ומתארים אותן בשרטוטים שניתן להבין בקלות גם ללא ידע בקוד או במידול.

למשל - וכולנו מכירים זאת היטב - לא תמיד אנחנו מבינים בדיוק למה התכוון איש הפרודקט בדרישה א' או ב'. האם איש הביזנס כיסה את כל המקרים החשובים במסמך הדרישות שלו? האם יש בכלל מסמך דרישות או שפתח ב-issue ב-Jira עם כותרת בלבד? וכמובן, ייתכן שבשלב מאוחר יותר מגלים שיש התנגשות בין דרישות מסוימות ואחרות לבין הפיתוח ואולי אפילו גם לדרישת הלקוח.

מודלים עוזרים לנו להתמודד עם אתגרים אלו בכך שהם מספקים לנו **"תמונת-על" של המערכת**. תמונה זו ניתנת לייצוג על ידי שרטוטים שכל הגורמים המעורבים בפיתוח יודעים לקרוא. כך כל הגורמים מבינים בדיוק את התנהגות המערכת המתוכננת, ויכולים להצביע על שגיאות ואי התאמות מול התנהגות המערכת הרצויה, כפי שהלקוח אמנהלי המוצר מבינים אותה.



“MBT היא שיטת בדיקות בה הטסטים נוצרים בצורה אוטומטית מתוך מודל המתאר את האספקטים הפונקציונליים של המערכת הנבדקת (ה-SUT)”

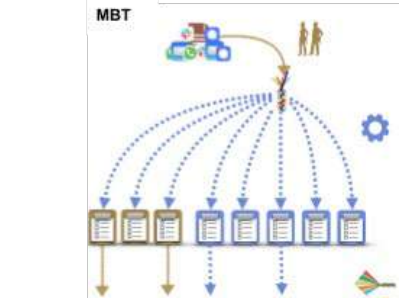
אז מה זה בעצם MBT?

MBT היא שיטת בדיקות בה הטסטים נוצרים בצורה אוטומטית מתוך מודל המתאר את האספקטים הפונקציונליים של המערכת הנבדקת (ה-SUT).

השוני המרכזי בין שיטות הבדיקה המסורתיות לבין אלו המבוססות מודל, נמצא בשלבי התכנון והיצירה של תסריטי בדיקה. אם עד עכשיו תכנון וביצוע של מקרי הבדיקה התבצע באופן ידני, על בסיס ניתוח של מסמכי דרישות, כעת יצירתם נעשית באופן אוטומטי, מתוך המודל.

כיוון שהמודל מייצג את דרישות המערכת העדכניות, הטסטים הנוצרים ממנו תמיד משקפים דרישות אלו. כך נוצר מצב בו הטסטים תמיד מעודכנים. בנוסף, ניתן למדוד כיסוי פונקציונלי של חבילות בדיקה בצורה מדויקת, וניתן להכווין את ייצור חבילות הבדיקה כך שיכסו פיז'רים אותם אנו מעוניינים לבדוק.

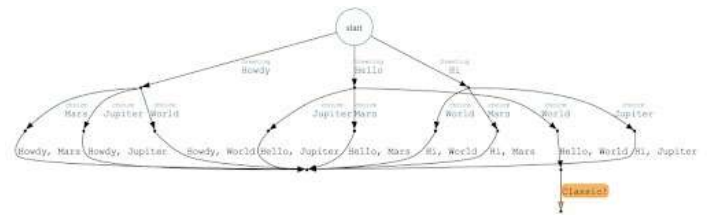
למשל, נניח כי אנו רוצים לבדוק אפליקציה בנקאית. ישנם סוגים שונים של משתמשים (נוער, מבוגרים, פנסיונרים),



סוגים שונים של חשבונות (עסקי, פרטי, השקעות) ומגננוני הזדהות שונים (זיהוי פנים, טביעת אצבע, סיסמא חד פעמית, וסיסמא קבועה). בשביל לסבך עוד טיפה - ולבטל את האופציה הפשוטה של מכפלה קרטזית בין האופציות האלו - נניח שחשבון עסקי יכול להיות רק למבוגר (לא לנוער ולא לפנסיונרים), שפנסיונרים לא עובדים עם סיסמא חד פעמית, ושחשבון עסקי דורש הזדהות ביומטרית כלשהי (זיהוי פנים או טביעת אצבע). המודל יכיל את כל השילובים האפשריים של הפרמטרים לעיל, ואת ההתנהגויות הרצויות של המערכת המוכתבות על ידי פרמטרים אלו (לדוגמא - בהזדהות עם סיסמא קבועה צריך להכניס שם משתמש וסיסמא, כאשר נוער נכנס למסך הראשי מציעים לו כרטיסים לטיילור סוויפט, וכד').

עכשיו, כשהמודל מוכן, ניתן לייצר חבילות בדיקה שונות. חבילת בדיקות מהירה תכלול טסטים בהם כל ערכי הפרמטרים מופיעים, אבל לא כל השילובים שלהם (נניח, יהיה לנו משתמש בוגר עם חשבון עסקי, לפני שחרור השקעות, אבל לא משתמש בוגר עם חשבון פרטי). לפני שחרור גרסה, נריץ טסטים מלאים יותר, בהתאם למגבלות הזמן והתקציב. אם בוצע שינוי לחשבונות נוער בלבד, נפיק חבילת בדיקות בה סוג המשתמש הוא תמיד "נוער", ושאר הפרמטרים משתנים. יצירת חבילת בדיקות כזו אורכת לרוב פחות מדקה¹.

אחד היתרונות הבולטים של מידול הוא הסיוע בזיהוי של באגים, טעויות או התנגשויות כבר בשלב כתיבת הדרישות במחזורי הפיתוח. כולם מדברים היום על Shift Left, פרדיגמת הזזת הבדיקות לשלב הפיתוח, המאפשרת זיהוי בעיות ופיתורן בשלב מוקדם יחסית של סבב הפיתוח.



```
//Base flow: choose a greeting, choose a planet, greet.

const GREETINGS = ["Hello", "Hi", "Howdy"];

bthread("Greeting activity",function() {
  let greeting = select("Greeting").from(GREETINGS);
  let planet = choose("Jupiter", "Mars", "World");
  request(bp.Event(`${greeting}, ${planet}`));
});

//When producing the classic "Hello World" greeting,
//note that the classic path was chosen.
bthread("Programmer", function(){
  waitfor(bp.Event("Hello, World"));
  Ctrl.doMark("Classic!");
});
```

במילים אחרות, המודל הוא נקודת ייחוס מרכזית בארגון ליצירת מקרי בדיקה, להרצה, או הערכה של המערכת. בצורה זו, כל הגורמים בארגון מבינים טוב יותר את מבנה המערכת, ואיך כל אחד מצפה שהיא תתנהג. השימוש במודל אחד מבטיח שכולם "על אותו הגל", מדברים באותה שפה למען מטרה משותפת - מוצרי תוכנה באיכות גבוהה.

אם כן, אנו מתחילים להבין את חשיבותו ואת כוחו של המודל בעולמות של תהליכי פיתוח תוכנה.

מודלים מפשטים את מורכבות המערכת על ידי אבסטרקציה של הפיצ'רים וההתנהגויות ההכרחיות, והם בעלי יכולת אדפטציה לאספקטים השונים של דרישות המערכת.

מה מודל צריך להכיל?

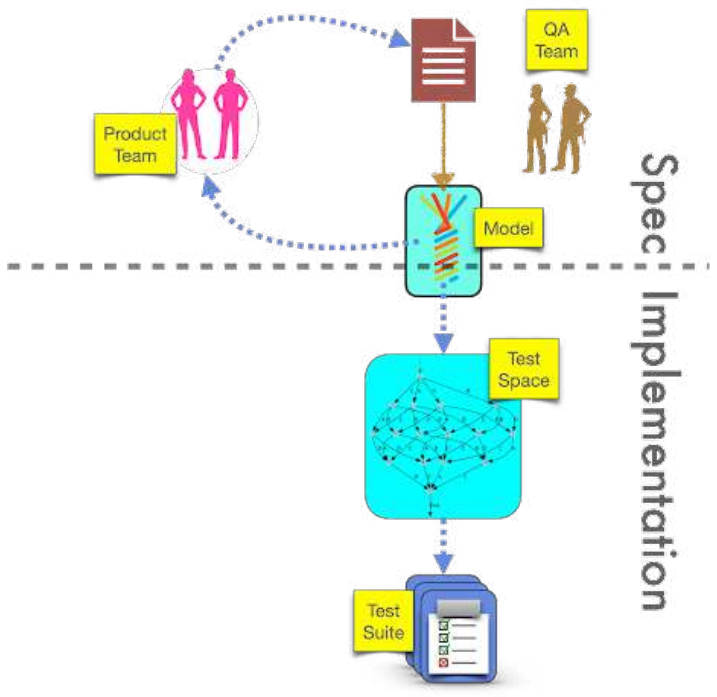
לכל מערכת יש את הטבע שלה והדרישות שלה. באופן כללי, נרצה שהמודל ידע ללכוד:

- התנהגויות פונקציונליות; ייצוג של רצף הפעולות, התלויות והמעברים בתוך המערכת.
- ניהול של שגיאות וחרגות. סנכרון בין רכיבים ומשאבים שונים וניהול של race conditions.
- הגבלות של משאבים שונים במערכת ושינויים ב-business flow.
- על המודל לתאר את מגוון התוצאות התקינות.
- התנהגות מול העולם החיצון; לקוח, פרטוקולים, זרמי דאטה, הודעות..

מהי רמת הפירוט הנדרשת?

נשאלת השאלה כמה מפורטים עלינו להיות בתיאור המודל על מנת שהוא יהיה יעיל. מסתבר שלא חייבים להיות מאד מפורטים. ניתן להתחיל מסט התחלתי מצומצם של דרישות הכרחיות ולהוסיף פירוט בהדרגה ככל שנדע טוב יותר מהם צרכי המערכת ואילוץ הבדיקה. בנוסף, רמת הפירוט של המודל לא חייבת להיות אחידה. למשל, אם אנחנו רוצים לבדוק מסך ספציפי במערכת, מודל הבדיקות יתאר את המסך בפירוט, אבל את הדרך בה המשתמש מגיע אל המסך הזה ברמה המינימלית ההכרחית.

1 נמדד עם כלים של Provengo על לפטופ MacBook Pro M1, עם מודל המכיל עשרות אלפי בדיקות.



מאחר והמודל נבנה ישירות מתוך הדרישות, הבודקים יכולים לזהות כשלים כמו הבנות שגויות, חוסר עקביות, או דרישות חסרות בשלב הראשוני ביותר של סבב הפיתוח - שלב האפיון והגדרת הדרישות העסקיות.

לדוגמה, נניח כי אנו בודקים שירות ענן של בנק המאפשר ללקוחות לקבל את המאזן הנוכחי בחשבונם. נניח רגע כי התקשורת לשירות היא דרך REST API. למערכת כזו ישנן מספר דרישות. ביניהן מופיעות שתי הדרישות הבאות:

1. אם המשתמש שלח מספר חשבון לא קיים, יש להחזיר הודעת שגיאה NOT FOUND 404
2. אם המשתמש שלח הזדהות לא נכונה, יש להחזיר הודעת שגיאה UNAUTHORIZED 401

אנשי ה-QA מוסיפים את שתי הדרישות למודל, יחד עם שאר הדרישות. כאשר המודל יבחן את מרחב הבדיקות האפשרי, הוא ימצא מקרה בו המשתמש שולח בקשה עם מספר חשבון לא קיים וגם עם הזדהות לא נכונה. במקרה כזה, המודל יצפה לתשובה שהיא גם 404 (בגלל דרישה 1) וגם 401 (בגלל דרישה 2). זה כמובן בלתי אפשרי, והמודל יציג שגיאה מתאימה, המצביעה על סתירה בין הדרישות. אנשי ה-QA יקחו את העניין לצוות האפיון שיעדכן את הדרישות.

“אחד היתרונות הבולטים של מידול הוא הסיוע בזיהוי של באגים, טעויות או התנגשויות כבר בשלב כתיבת הדרישות במחזורי הפיתוח”

אגב - בדוגמה לעיל הטסטים בכלל לא נגעו במערכת הנבדקת. בדיקות מהסוג הזה יכולות להתבצע לפני שאנשי הפיתוח כתבו אפילו שורת קוד אחת!

שימוש ב-MBT מאפשר לתהליך ה-QA להתחיל כבר בשלב הדרישות. כמוכן, שרק מעצם קבלת פידבק בשלב מוקדם זה אנו מאפשרים העברת דרישות הדוקות ואיכותיות יותר לפיתוח. ברור כי דרישות טובות ומסודרות חוסכות משמעותית בעלויות ובזמן הפיתוח, וב-time to market.

כך, שימוש ב-MBT מאפשר לשפר את איכות המוצר, ולהוריד את זמן ואת עלויות הפיתוח - שלושת ה-KPIs לפיהם, על פי רוב, צוותי פיתוח נבחנים.

MBT, אוטומציה, ובדיקות ידניות

בדיקת דרישות היא יכולת ייחודית לגישת ה-MBT, אבל כמוכן שגם ניתן לייצר בעזרתה בדיקות רגילות. ייצור בדיקות כאלו יתבצע בדרך כלל על ידי הוספת הוראות ביצוע למודל קיים. למשל, נניח שיש לנו מודל המתאר מסע לקוח באתר בנק. המודל הבסיסי יתאר תהליך בסטגון "המשתמש פותח את מסך הזדהות, משם עובר לעמוד הראשי, ומשם עובר למסך מצב חשבון". על מנת לאפשר אוטומציה ישירות מהמודל, נוכל להוסיף קוד סלניום שירוץ בכל מסך. אם נרצה לעבור ליצירת ספר בדיקות ידני, נחליף את שכבת ההוראות של הסלניום בשכבה אחרת, המפרטת צעדים באופן המתאים לבודק ידני. כאשר הבנק יציג אפליקציה שמקבילה לאתר, נוכל להשתמש במודל הבסיסי שוב, אבל הפעם להחליף את שכבת ההוראות הסלניום בשכבה עם הוראות של Appium.

בשיטות אלו ניתן ליצור חבילות חדשות של בדיקות אוטומציה, או חבילות של בדיקות ידניות. אבל מה עושה מי שכבר יש לו תשתיות בדיקה עשירות? לא חבל לזרוק?

החדשות הטובות הן שלא צריך לזרוק תשתיות קיימות - ניתן להשתמש ב-MBT על מנת לייצר תסריטי בדיקה המתבססים על תשתיות אלו. כך, במקום לתרגם חבילות בדיקות לספר בדיקות ידני, ניתן לתרגם אותה לקבצים ב-Python או Java/JUnit, או לכל פורמט מבוסס טקסט אחר. כולל, כמובן, קבצי JSON ו-XML. שילוב כזה מאפשר לצוות ה-QA גמישות רבה בבניית המודל, כיוון שהם יכולים להחליט אילו פעולות צריכות להיות מפורטות כחלק מהמודל עצמו, ואילו פעולות יכולות להיכנס ל-"קופסאות השחורות" של ספריית האוטומציה. בנוסף, גישה זו מאפשרת ניצול מיטבי של ספריית האוטומציה, כיוון שייצור תסריטי בדיקה המתבססים עליה מתבצע באופן אוטומטי.

MBT ופיתוח אג'ילי

פיתוח אג'ילי מאפשר - מעודד אפילו - שינויים תכופים במערכת המפותחת, על מנת להתאים אותה לצרכים המשתנים של הלקוח. שינויים אלו יוצרים דילמה מתמדת בקרב צוותי ה-R&D: אבטחת איכות לעומת תחזוקה. אם נכתוב הרבה טסטים נקבל כיסוי בדיקות יסודי שיבטיח איכות גבוהה. מצד שני, כמות גדולה של טסטים פוגעת באג'יליות של הצוות, כיוון שכל שינוי בתוכנה דורש בדיקה, עדכון, וכתיבה מחדש של כל הטסטים הקיימים - פעולות הדורשות זמן ועבודה רבה. כאשר מדובר בטסטים אוטומטיים הבעיה מתחדדת. כמוכן, אם אנו בוחרים לקצץ בכמות הבדיקות בעיית התחזוקה נפתרת, אולם אנו מסתכנים בפגיעה באיכות המוצר עקב כיסוי בדיקות נמוך.

כך למעשה נולדו שיטות הפירמידה, לפיהן בדיקות מסוג E2E, שהן הבדיקות הכי חשובות, נמצאות בראש הפירמידה ואותן מבצעים הכי מעט.

עם MBT, ברגע שכבר יש לנו מודל בסיסי לעבוד מולו, כל דרישה שנרצה להוריד, להוסיף או לשנות, תדרוש עדכון למודל - לא לטסטים עצמם. אחרי העדכון, נמחק את הטסטים הקיימים, ונייצר באופן אוטומטי חבילות בדיקה חדשות מתוך המודל המעודכן. כך, שינוי דרישות שבשיטות בדיקה רגילות דורש עדכון של אלפי תסריטי בדיקה, ידרוש שינוי בודד במודל כאשר משתמשים בגישה של MBT.

למעשה, MBT מאפשר לנו אוטומציה של יצירת תסריטי בדיקה בודדים ושל חבילות בדיקה המשלבות תסריטי בדיקה מרובים. כך, לא רק שנחסך לנו המון זמן של תכנון, הוצאה לפועל ותחזוקה ידנית, אלא גם מובטח לנו שחבילות הבדיקה יניבו כיסוי פונקציונלי גבוה יותר וישקפו באופן מעודכן ונכון את דרישות המערכת הנבדקת.

אז איך MBT מועיל לאנשי הבדיקות

תפקידו של הבודק משתנה מעט ומשתפר הרבה. הוא הופך להיות מעין אנליסט, המתמקד בהבנת הדרישות, הלוגיקה, והאפיון של המערכת - במקום ביצירת הסקריפטים ותחזוקתם. מכיוון ש-MBT מאפשר לבצע QA כבר בשלב האפיון, ולוודא את נכונות ואיכות הדרישות עצמן, הבודק עובד צמוד עם מחלקת ה-Product, אולי אפילו לפני כניסתם של המתכנתים לתמונה.



לסיכום

הצגנו את גישת הבדיקות מבוססת המודלים (MBT). ראינו כי היא מאפשרת למשתמשים בה להנות ממספר יתרונות רב. בין השאר שיפור איכות העבודה והתוצרים, שיפור התקשורת בין ארגונים שונים ובין קבוצות הפיתוח השונות, הוזלת עלויות הפיתוח, והגברת מהירות הפיתוח.

- מיפוי כלל ההתנהגויות במערכת והצגתם באופן ויזואלי שאנשי ה-business והפרודקט יוכלו להבין באופן אינטואיטיבי וקל (Validation)
- אפשרות לבדוק איכות כבר בשלב איסוף הדרישות, כלומר מציאת באגים ודרישות סותרות לפני שבניית המערכת הנבדקת עצמה החלה (Verification)
- הפקה אוטומטית של חבילות בדיקה אופטימליות בסנכרון מלא לדרישות המערכת, כך שהבדיקות אכן בודקות את מה שהצד העסקי ביקש לבדוק
- מעבר קל ואינטואיטיבי בין יצירת בדיקות אוטומטיות לבין יצירת ספרי בדיקות ידניים
- תחזוקה קלה ופשוטה של חבילות בדיקה

היתרונות

מאמצי התחזוקה של הסקריפטים יורדים למינימום - המודל הוא ייצוג של הדרישות, ותסריטי בדיקה מיוצרים אוטומטית מתוך המודל. לכן, ברגע שחל שינוי בדרישות, הבודק יעדכן את המודל וזה יג'נרט לו בצורה אוטומטית את התסריטים החדשים.

בהתאם, קצב יצירת תסריטי הבדיקה (האוטומטים והידניים) יגדל מאוד. מכיוון שיש הפרדה ברורה בין תסריטי הבדיקה למימושם, קוד המודל יהיה "קוד נקי" באופן דיפולטיבי (clean code).

מה הקאץ'?

השימוש מצריך שינוי חשיבתי; עד היום בבדיקות מבוססות תסריטים היינו מתמקדים כל פעם בריצה נכונה אחת של המערכת הנבדקת. עם השימוש ב-MBT עלינו לסגל חשיבה הבוחנת את התנהגות המערכת כמכלול (כאמור - לא ברמת פירוט מלאה). כמו כן, נדרשת השקעה בהכשרות וכלים מתאימים ובביצוע אינטגרציה למערכות הקיימות.

מדוע MBT לא תפס עד עכשיו?

עולם המידול החל להתפתח לכיוונים של בדיקות כבר לקראת סוף שנות ה-90, עם לא מעט כלים שונים לאורך השנים שזכו להצלחה חלקית. מאוד.

כל היתרונות של בדיקות מבוססות מודל תלויים, איך לא - בטיבו של המודל. היכולת לייצר **מודל טוב** היא כשלעצמה דבר בכלל לא פשוט. בדומה לבניית STD מוצלח, כדי לייצר מודל מוצלח ויעיל דרושה קודם כל מומחיות והבנה מעמיקה בתחום בו המערכת עוסקת. עלינו לעדכן אותו בצורה איטרטיבית, בתגובה לדאטה חדש, לפידבק, ולשינוי בדרישות או בסיפורי המשתמשים. כל אלו דורשים זמן, כסף ולופ של פידבקים בין כל בעלי העניין. יש גם קושי במציאת רמת ההפשטה הנכונה על מנת לייצר מודל שימושי של מערכת מורכבת מהעולם האמיתי. נוסף על כך, ישנם קשיים בלכידת התנהגויות לא דטרמיניסטיות וכן בתחזוקה של המודל. זאת מכיוון ששמירה על בהירות המודל נעשית קשה יותר ככל שמורכבות המערכת גדלה.

אז למה דווקא עכשיו?

ב-2010 פרסם צוות חוקרים ממכון וייצמן - דוד הראל, אסף מרון, וגרא וייס - פרדיגמה חדשה ליצירת מודלים. הם קראו לשיטה Behavioral Programming - BP, כלומר **תכנות התנהגותי**, והיא **מותאמת לאופן בו אנשים בדרך כלל מתארים התנהגות של מערכת**. פרדיגמה זו מאפשרת ליצור מודלי בדיקה בצורה הרבה יותר אינטואיטיבית ונוחה בהשוואה לשיטות שקדמו לה.

בגדול, מודל BP מורכב מהרבה התנהגויות קטנות ופשוטות. כל התנהגות כזו מייצגת דרישה או use case המופיע במסמך הדרישות, בעזרת רצף אירועים שהמערכת צריכה, יכולה, או אסור לה להוציא לפועל. התנהגות יכולה להיות עצמאית לגמרי, או מאוד תלויה בהתנהגויות אחרות, או בכל מקום ביניהם. שילוב כל ההתנהגויות הקטנות ביחד יוצר מודל שלם של ההתנהגות הרצויה של המערכת הנבדקת, שהיא - בדרך כלל - התנהגות מורכבת מאוד.

מאז 2010 נעשו שינויים ושיפורים רבים לפרדיגמה זו, ופותחו טכנולוגיות וכלים לעבודה עם מודלים שנבנו על פיה. כיום, באמצעות כלי המידול והבדיקות של חברת Provengo, בודקי תוכנה בעלי ידע בסקריפטינג או תכנות בסיסי יכולים להצטרף לעולם ה-MBT בצורה קלה ואינטואיטיבית. ניתן לפתח ולבדוק מודלים, לייצר דיאגרמות התנהגות, לבנות חבילות בדיקה אופטימליות, ולהריץ את החבילות האלו ידנית, באוטומציה, או על ידי יצירת קוד למערכות בדיקה קיימות.

HISTORY OF DEBUGGING MONKEYUSER.COM





קארין בוחבוט



שמי קארין בוחבוט, ראש צוות בדיקות ואוטומציה בחברת פריון. אני עובדת בפריון קרוב לתשע שנים, שמונה מתוכם כראש צוות בדיקות. לאחר כשנה, נכנסנו לעולמות האוטומציה ואני המשכתי לנהל את הצוות. בתפקיד הנוכחי שלי, אני מנהלת ומנחה את הצוות, מפתחת אסטרטגיות בדיקות, ומובילה את התהליכים שלנו כדי להבטיח את איכות המוצרים שלנו וביצועים מיטביים. כיום, מעבר לתפקיד הפורמלי, אני מובילה את גילדת ה-QA בארגון, שבה חברים כל אנשי הבדיקות מכל היחידות, בכדי למקסם העשרה והעברת ידע בין היחידות. בשנים האחרונות אנו מתמקעים בטכניקות אוטומציה מתקדמות, שמטרתן להקל על התהליך ולמקסם את היעילות של הבדיקות אותן אנו מבצעים



במה עוסקת הקבוצה וכיצד היא בנויה?

פריון נטוורק (סימול: PERI), היא חברת טכנולוגיה המספקת פתרונות טכנולוגיים לפרסום דיגיטלי בערוצי חיפוש, רשתות חברתיות, אתרי תוכן, וידאו ופרסום בטלוויזיות חכמות.

כחלק מהיצע הפתרונות הטכנולוגיים, מספקת החברה פתרונות מונטיזציה בחיפוש, למפרסמים ובעלי אפליקציות המעוניינים לייצר רווחים מאתרי תוכן, מנועי חיפוש, דפדפנים, הרחבות ב-chrome & firefox ובאפליקציות אחרות. אני חלק מחטיבת החיפוש של פריון, החטיבה מונה כ-70 אנשים, ומורכבת מקבוצות פיתוח, דאטה, מדיה, מכירות, עיצוב ומנהלי מוצר. צוות האוטומציה הוא חלק מקבוצת הפיתוח. כל קבוצה מורכבת ממפתחים, ראש צוות פיתוח ואיש אוטומציה. איש האוטומציה מוטמע בתוך הצוות, והוא חלק חשוב ואינטגרלי החל משלב התכנון ועד לשלב הביצוע. בנוסף, הצוות מתפקד כצוות אוטומציה, ומספק פתרונות אוטומטיים, הן ברמת ה-API והן ברמת ה-UI.

איך נראה יום העבודה שלך?

אנחנו עובדים בשיטה של scrum בספרינטים של שבועיים. כל יום עבודה מתחיל בשיחת דיילי יומית, בה כל איש צוות מספר על התקדמותו והאתגרים שבהם נתקל, מפרט על מה הוא עובד, ומתייחס למה שעשה. בכך, אנו מבטיחים סנכרון מוחלט בין חברי הצוות והבנה מדויקת של המצב בכל רגע נתון. התהליך מבטיח שכולם מודעים לאתגרים הקיימים ופועלים ביעילות מרבית כדי לפתור אותם. במסגרת תפקידי אני מנהלת 4 אנשים בקבוצות שונות, ואני מחלקת את הזמן שלי בין המשימות.

באילו אתגרים נתקלים הבודקים שלך?

האתגר הגדול ביותר שהבודקים נתקלים בו, הוא היכולת להיות הגורם המקשר בין כל ה"שחקנים" הרלוונטיים במערכת. איש הבדיקות מעורב החל משלב הפיתוח, ועד לשלב הביצוע ולשלב ההעלאה לייצור. לכן, הוא תמיד צריך להיות מסונכרן, לעקוב אחרי שינויים, ולאשר את בדיקות הקצה (E2E) של המוצר.

באילו אתגרים ניהוליים הנך נתקלת?

- ניהול, השקעה ופיתוח של האנשים בצוות. פיתוח והכשרת הצוות, הבנת היכולות הפרטיות של כל חבר בצוות, ויצירת סביבת עבודה תומכת ומעוררת מוטיבציה.
- ניהול זמן- מציאת האיזון הנכון בין פרויקטים מרובים וניהול תוכניות הפעולה למניעת עיכובים בזמנים הוא אתגר תמידי.
- תקשורת בין הצוותים השונים- תיקון התקשורת בין הצוותים השונים בחברה כדי לקדם פרויקטים בצורה מהירה ויעילה. הבנת הצרכים של כל צוות והפקת תוצאות רצויות.

כיצד הנכם פועלים להעשרת הידע של הבודקים?

אנו מקנים חשיבות רבה לפיתוח מקצועי ורמת המומחיות בצוות. על מנת לקדם את הידע הטכני והמקצועי שלנו, אנו מביאים מומחים חיצוניים לחברה להקניית קורסים מקצועיים בתחום, משלבים מיטאפים, ומציינים בכל חודש זמן מיוחד לגילדה משולבת. במהלך כל גילדה, כל אחד מצוות בדיקות מספק ידע עמוק וניסיון בכלי

מה היית ממליצה לבודק שנמצא בתחילת דרכו?

תחילת הדרך בתחום הבדיקות היא הזדמנות נהדרת להתעשר בידע ולהתפתח מקצה לקצה. המלצתי היא להיות סקרן, לקחת חלק בקורסים וסדנאות מקצועיות, ולהתרכז בלמידת כלים וטכניקות חדשות בעולם הבדיקות. חשוב להיות מוכן לשאול שאלות, לבחון פרויקטים קיימים ולהבין מהם, ולשתף פעולה עם שאר חברי הצוות. אל תפחד לטעות, מטעויות אפשר ללמוד הכל!! תחום הבדיקות הוא תחום דינמי שכל הזמן מתרחש בו משהו חדש, אז חשוב להיות רגיש לשינויים ולהתעדכן באופן יסודי. בנוסף, רכש ניסיון ממגוון פרויקטים ונסה לעבוד עם מגוון טכנולוגיות וכלים, זה יפתח את יכולותיך ויחזק את ניסיוןך. והכי חשוב ליזכר, שניתן ללמוד כל דבר, חשוב להיות עם ראש פתוח.

פספורט קבוצתי Perion

סוג המוצר הנבדק: ווב, דסקטופ, סרביסים.

גודל קבוצת הבדיקות: הצוות מורכב מאנשים בעלי וותק של בין שנה ל-10 שנים.

מבנה הקבוצה: צוות האוטומציה מכיל 4 אנשים, כל אחד הוא חלק אינטגרלי מקבוצת הפיתוח.

סוגי בדיקות: בדיקות ממשק משתמש הן ברמת API, פונקציונאליות, נסיגה ושפיות.

שיטת עבודה: אנו עובדים בשיטת אג'יל סקראם בספרנטים של שבועיים.

כמות המוצרים הנבדקים וקצב שחרור

גרסאות: יש לנו מגוון רחב של מוצרים אשר קצב שחרור הגרסאות הינו אחת לשבוע.



המראיין: ניצן גולדנברג
מזה 8 שנים בתחום בדיקות התוכנה, בתפקיד נוכחי מהנדס בדיקות בכיר בחברת Seatgeek. מנהל קבוצת ה-AB של ארגון ITCB המוביל הראשי של קבוצת המיטאפ TestIL ומרצה בקורסים לבודקי תוכנה





מדדי בדיקות תוכנה Software Test Metrics



קובי יונסי

בעל תואר ראשון מאוניברסיטת תל אביב בתחום הלוגיסטיקה והכלכלה. מרצה מוביל במכללות הגדולות בישראל וראש החוג בטכניון בתעשייה מרצה הבית של חברות מיקור החוץ הגדולות במשק. מייסד ומקים את המרכז המוביל לבדיקות תוכנה. מלמד אנשים כיצד לחשוב יצירתי ולשפר את מיומנויות הבדיקה.



קלישאה ידועה אומרת שמה שלא נמדד לא נעשה וזאת מאחר שבאופן טבעי כשאנחנו מודדים משימה יש נטייה טבעית להעריך אותה ביתר או בחסר או פשוט להתעלם ממנה.

"כאשר מטרת התכנון ומהותו היא לתרגם מחשבות מהטווח הארוך לפעילויות מתקנות בטווח הקצר בכדי שנהיה ליוזמים ולא למגיבים"

חסכון בזמן ומשאבים: השימוש במדדים מאפשר לזהות במהירות תקלות ובאגים, ובכך לחסוך בזמן ובמשאבים. זה יכול להפחית את העלויות ולקצר את מחזור הפיתוח.

שיפור מתודולוגיות: מדדי בדיקות עוזרים לצוותי הבדיקה להבין אילו מתודולוגיות עובדות בצורה טובה ואילו דורשות שיפור.

התאמה לצרכי הלקוח: המדדים מסייעים להבטיח שהתוכנה תתאים לצרכי הלקוחות ותענה על הציפיות שלהם, מה שמעלה את הסיכוי לשחרור מוצר מוצלח החוצה.

בהיבט הבדיקות ישנם חמישה ממדים שדרכם נבצע מעקב לבדיקות:

1. **סיכוני מוצר,** מדדים של סיכוני מוצר יכולים לכלול:
 - 1.1 אחוז הסיכון מהמוצר ביחס למקרי בדיקה שעברו בהצלחה.
 - 1.2 אחוז הסיכון מהמוצר מול מקרי בדיקה שנכשלו.
 - 1.3 אחוז הסיכון מהמוצר מול בדיקות שטרם נבדקו.
 - 1.4 אחוז הסיכון ביחס לסיכון הכללי מהמוצר.

* הפורמולה הבסיסית לחישוב סיכון תהיה:
אחוז הסיכון = תדירות הסיכון X השפעת הסיכון

2. **תקלות,** מדדים של תקלות יכולים לכלול:
 - 2.1 כמות תקלות שדווחו מול כמות תקלות שנפתרו.
 - 2.2 כמות תקלות מול רכיבים שנבדקו.
 - 2.3 כמות תקלות ביחס לגרסה, מהדורה של המוצר.
 - 2.4 כמות תקלות וכמה מהן נדחו או שהן כפולות (Duplicated).
 - 2.5 כמות תקלות וכמה מהן שנפתרו ויצרו תקלות חדשות. (נקרא גם: Daughter Bugs).
3. **בדיקות,** מדדים של בדיקות יכולים לכלול:
 - 3.1 כמות הבדיקות שתוכננו מול בדיקות שנכתבו בפועל/ שהורצו בהצלחה/ שנכשלו/ שלא נבדקו.
 - 3.2 סטטוס של בדיקות גרסיה ובדיקות אימות.
 - 3.3 כמות שעות יומית שתוכננו לבדיקות וכמות השעות היומית ששימשו אותנו בפועל.
4. **כיסוי,** מדדים של כיסוי יכולים לכלול:
 - 4.1 מדד כיסוי הדרישות שהשגנו.

מדדי בדיקות תוכנה הם אינדיקטורים הניתנים לכימות וקשורים להתקדמות תהליך בדיקת התוכנה בהיבטים שונים של הפרויקט. מטרת מדדי בדיקות התוכנה היא להגביר את היעילות והאפקטיביות של תהליכי הבדיקות תוך סיוע בקבלת החלטות טובות יותר לבדיקות עתידיות על ידי מתן נתונים מדויקים על תהליך הבדיקה שהתבצע בעבר ומתבצע בהווה.

ניתן לסווג מדדי בדיקות לכמה קטגוריות:

- **מדדי פרויקט** – שמטרתם לוודא את התקדמות הפרויקט לקראת קריטריון היציאה כמו למשל אחוז מקרי בדיקה שהורצו, עברו בהצלחה או נכשלו.
- **מדדי מוצר** – שמזהים תרומה למוצר כמו למשל כמה כיסוי של מקרי בדיקה השגנו וכמה ממקרי הבדיקה שהורצו עברו בהצלחה.
- **מדדי תהליכים** - מזהים התקדמות בבדיקות או בפיתוח כמו למשל אחוז התקלות שהתגלו ביחס לבדיקות שבוצעו.
- **מדדי אנשים** – מאפשרים לזהות ביצוע של בעלי תפקיד בפרויקט, כמו למשל כמות הבדיקות שהורצו בפועל על ידי צוות הבדיקות ביחס ללוח הזמנים המצופה מהם.

חשוב לציין שישנם מדדים שיכולים לסייע לזהות התקדמות בכמה קטגוריות יחדיו, כמו למשל גרף שמזהה כמות דיווחי תקלות יומי מאפשר לנו לדעת מהי איכות המוצר בשלב נתון ומכאן להוות מדד למוצר אך במקביל מאפשר לקבל תמונה ביחס למצבו של תהליך הבדיקות באופן כללי, איפה אנחנו עומדים ומה נותר לנו לבדוק.

כבודקים מה שבדרך כלל יעניין אותנו הם המדדים ביחס לתהליך הבדיקות אותם נצטרך להציג מעת לעת בישיבות צוות פורמליות ולא פורמליות מול בעלי העניין בפרויקט (אנשי הפיתוח, הלקוחות, מנהלי המוצר ושחקנים נוספים שלוקחים חלק בתהליך)

שימוש במדדים מסייע לבודקים להציג תוצאות באופן עקבי ומסודר, לעקוב אחר קצב ההתקדמות, ולזהות מראש נקודות חולשה שיכולות להתפתח לצווארי בקבוק בפרויקט אם לא יקבלו מענה בזמן.

"כבודקים מה שבדרך כלל יעניין אותנו הם המדדים ביחס לתהליך הבדיקות"

לפיכך מדדי בדיקות הם חלק בלתי נפרד בשלב תכנון הבדיקות (מסמכי STP) וגם בשלב תהליך הבדיקות עד לסגירתם ומטרתם לסייע לאנשי הבדיקות לתעד את סטטוס הבדיקות בכל שלב מאבני הדרך של הפרויקט.

כאשר מטרת התכנון ומהותו היא לתרגם מחשבות מהטווח הארוך לפעילויות מתקנות בטווח הקצר בכדי שנהיה ליוזמים ולא למגיבים.

מדדי בדיקות בתהליך בדיקות הם חשובים ביותר, מכיוון שהם מסייעים לצוותי הבדיקה לקבוע את איכות התוכנה ולוודא שהיא תואמת את התקנים וציפיות הלקוחות. הנה כמה סיבות לחשיבות מדדי בדיקות בתהליך הבדיקות

איכות התוכנה: מדדי בדיקות מאפשרים להעריך את איכות התוכנה. הם מסייעים לזהות בעיות, באגים ותקלות, ובכך להעריך האם פיתוח התוכנה עונה על ציפיות הלקוחות.

שימושיות: המדדים מאפשרים לצוותי הבדיקה לקבוע אם התוכנה ישימה וקלה לשימוש, ובכך לשפר את חוויית המשתמש.

אמינות: מדדי בדיקות מסייעים לוודא שהתוכנה פועלת באופן אמיין ותקין ושהיא לא גורמת לכשלים או לתקלות בסביבה הפועלת.



היציאה ומאפשר לנו להחליט אם המערכת תימסר או לא תימסר ללקוח. לדוגמה אם 40% מהבאגים הם חמורים, זה יחס שאינו סביר מבחינת איכות תוכנה שתימסר ללקוח ולכן ההחלטה הצפויה תהיה שלא למסור את המוצר.

שביעות רצון של הלקוח מדד זה משמש למדוד את שביעות רצון הלקוחות מהתוכנה. זהו מדד חשוב מאוד, שמראה כמה התוכנה עונה על ציפיות המשתמשים והאם אנחנו בכיוון של GO או NOT GO מבחינתו של משתמש הקצה הסופי.

למדדי הבדיקות יש מחזור חיים שכולל:

אנליזה – שלב הראשון בו נזהה מה הם המדדים שחשובים לנו ונרצה להשתמש בהם.

תקשורת – שלב שבו ניידע את כל מי שקשור לפרויקט לגבי הדרישה שלנו למדדים אלו. כמו כן נלמד את צוות הבדיקות לגבי הנתונים ומקורות הנתונים שאותם נאסוף לצורך עיבוד שוטף. לפעמים צריך לפתח קוד או לקסטם את מערכת ניהול הבדיקות בכדי שתפיק לנו את הערכים הנמדדים בצורה נוחה למעקב ולהצגה.



הערכה – ניתוח ועיבוד הנתונים לכדי חישוב שיאפשר לנו לקבל תמונת מצב מדויקת ועדכנית, ולפי כך נוכל לדעת האם אנחנו רצים וצמודים לתוכנית או שיש צורך בפעולות תיקון.

דיווח – הפצת דוח התקדמות הבדיקות על בסיס המדדים לבעלי העניין וקבלת משוב חוזר מהם לגבי המשך הפעילות בפגישות שנקיים לאורך תהליך הבדיקות.

לסיכום, מדדי בדיקות בתהליך בדיקות הם כלי ניהולי ממדרגה ראשונה והיות שכן, הם משמשים אותנו גם למעקב אחרי התקדמות הבדיקות במהלך הפרויקט ולכן ניתנת למדדים אלו חשיבות גבוהה באפשרות לסייע לצוותי הבדיקה להבטיח את איכות התוכנה, למנוע בעיות פוטנציאליות ולוודא שהתוכנה תואמת את התקנים וציפיות המשתמשים.

יש נושאים ומושגים נוספים שהייתם רוצים לקרוא עליהם? תשלחו מייל ואשמח לכתוב kobi757@gmail.com



4.2. מדד כיסוי הסיכונים שאליו הגענו שמצביע על רמת הסיכון שנותרה מהמערכת.

4.3. כיסוי תצורה וסביבה של המערכת כמו למשל כמות ההתקנות שהוכנסו בפלטפורמות שונות ביחס לתכנון המקורי.

4.4. כיסוי קוד שבוצע.

5. **בטחון, מדדים** של ביטחון לא ניתן לכמת במספרים ולכן נהוג בחלק זה לבצע סקרים מול בודקי התוכנה ביחס לתחושה שלהם מהמוצר ומרמת האיכות בזמן שבדקו את המוצר.

כמו כן נהוג לבקש חוות דעת מפורטת בשלב בדיקות הקבלה ממשתמשי הקצה הפוטנציאליים של המוצר בין אם מדובר בבדיקות ביטא או בדיקות אלפא.

מעבר למדדים אלו, ישנם מספר מדדים שיכולים בנוסף להוסיף לתודעת הבדיקות בארגון וללמד אותנו ואת מנהלי הפרויקט על קצב התקדמות הבדיקות ובשלותו של המוצר.

המרכזיים שבהם כוללים למשל:

עמידה בלוח הזמנים, המטרה העיקרית של עמידה בלוח היא לקבוע מהו ההפרש בין זמן הביצוע הצפוי לזה שבפועל כך שנוכל לבצע תיקונים בזמן ולהשקיע משאבים בכדי למנוע איחור צפוי בזמן עלייה לאוויר. פעולות מתקנות אלו הן חלק מבקרת הבדיקות (Test Control)

שיעור מציאת התקלות, יכול ללמד אותנו על יעילות צוות הבדיקות ועל טיב המערכת והקוד שפותח על ידי המפתחים.

ניתן לחשב מדד זה לפי גרסה, לפי אחוז יומי, שבועי, חודשי או מול מודול במערכת.

יעילות מקרי הבדיקה, מדד זה מאפשר לנו לקבוע אילו מקרי בדיקה הניבו תקלות בשיעורים גבוהים יותר ואז נוכל להשתמש במקרים אלו בגרסאות הבאות ולהעניק להם תיעדוף גבוה יותר כמקרי בדיקה עם אחוז גבוה יותר למציאת תקלות.

כדי לממש זאת ניתן להשתמש בפורמולה הבאה:

= **Test Case Effectiveness**

$$(Number\ of\ defects\ detected / Number\ of\ test\ cases\ run) \times 100$$

כיסוי קוד, Code Coverage מודד את החלקים בקוד שנבדקו כלומר היחס בין הקוד שנבדק לבין כלל הקוד שנכתב. המטרה היא לוודא שכמה שיותר מהקוד נבדק, כך שלא תהיה חסרות בבדיקה שעשויה להוביל לבעיות בפרקי התוכנה שלא נבדקו.

פשטות קוד, Code Complexity הפשטות של קוד הוא מדד שמשקף עד כמה הקוד מסובך. פשטות קוד זה קוד שנכתב בצורה מודולרית מחלקים קטנים ופשוטים ואין בו יותר מידי הסתעפויות ולכן קוד פשוט יותר נוטה להיות יותר קריא ופחות סביר שבו יהיו בעיות.

Defect Density מדד זה מציין את מספר התקלות שהתגלו בקוד ליחידה של מידות כמו שורות קוד, פונקציות, או יחידות אחרות. ערכים נמוכים במדד זה יכולים להצביע על איכות טובה של התוכנה.

זמן ממוצע בין כשלים, Mean Time Between Failures משמש לחשב את הזמן הממוצע בין כשלים בתוכנה. זמן זה משמש להערכת היציבות ואמינות התוכנה ומשקף את רמת הביטחון ממנה.

חומרה של באגים, Bugs Severity מדד זה משמש לסווג באגים לפי חומרתם ומתייחס ליחס בין מספר הבאגים החמורים לכלל הבאגים שנמצאו ובכך ניתן לתעדף את הטיפול בהם. באגים עם חומרה גבוהה יעלו לראש התור בתהליך התיקון ויקבלו עדיפות על פני באגים אחרים. מעבר לכך מדד זה מצוי במקרים רבים בקריטריוני



מה הן בדיקות רגרסיה

הרגרסיה בעברית משמעה **נסיגה**. בדיקות רגרסיה הן בעלות פוקוס ותשומת לב שונה לגמרי מבדיקות רגילות. בדרך כלל הבדיקות מתבצעות על מנת לוודא שאכן פיתוחים חדשים ותיקוני באגים עונים על הדרישות שהוגדרו עבורם.

לא כן הוא בבדיקות רגרסיה. כאן הגישה שונה, בדיקות אלה באות לוודא כיצד השינויים שנעשו בקוד של המוצר, בין בשילוב פיצ'ר חדש ובין בתיקון באגים – השפיעו על האיכות של המוצר. האם האיכות של המוצר **נסוגה**, דבר שיתבטא בפגיעה בחלקים ופיצ'רים קיימים כבר ואפילו באגים חדשים שלא היו קיימים אפילו בשלבי פיתוח מוקדמים יותר של פיצ'רים ישנים. או שמא במקרה הטוב – ההשתלה של הפיצ'רים החדשים והתיקונים עברה בהצלחה ללא תופעות לוואי ודחיה.

מצבים משתנים מצריכים גישות שונות

בדיקות רגרסיה משתנות מאוד בעיקר בהיקפן בהתאם למצבים שונים. לדוגמא, ככל שמוצר יותר מונוליטי, היינו, מקשה אחת של קוד, שנבנתה באופן לא מספיק מאורגן, שלצערנו זו לא תופעה נדירה כלל – נדרשת גישה יותר מחמירה, הדורשת היקף בדיקות רחב מאוד שכולל לעיתים רבות את כל המוצר. שהרי, מכיוון שהקוד של המוצר הוא כעין פקעת חוטים - אינך יודע מה ייקרה ומה יזוז אם תמשוך חוט מסוים או אפילו כמה.

זהו למעשה תהליך יקר ושחוק, מכיוון שהוא דורש המון זמן, משאבים ומאמצים של צוות הבדיקות לביצוע מחודש של תכולת הבדיקות כולה.

"הבדיקות מתבצעות על מנת לוודא שאכן פיתוחים חדשים ותיקוני באגים עונים על הדרישות שהוגדרו עבורם"

מאידך, כאשר מוצר מתוכנן טוב יותר ברמה הארכיטקטית, למשל מאורגן בצורה של מיקרו סרוויסים, מבוצר יותר בקוד, כך שלכל חלק בקוד יש מידה גדולה יותר של עצמאות - כך הסכנה ששינוי שנעשה יגרום לנזק מרחיק טווח במוצר קטנה בהרבה, וממילא נגזר מזה שתכולת הבדיקות לטובת שינויים אלו תהיה הרבה יותר מתוכננת ומצומצמת, מהירה וזולה יותר. כמובן שצוות הבדיקות יישחק פחות.

שתי הדוגמאות שלעיל משקפות שתי נקודות קיצון. כמובן שיכולים להיות הרבה מאוד מצבי ביניים, שרק מי שמכיר את המוצר היטב, יכול להתוות את האיזון הנכון בין שתי הגישות על מנת ליצור את החבילה הנכונה והמתאימה ביותר לבדיקות רגרסיה של המוצר.

עוד פקטור חשוב מאוד שצריך לדעתי להילקח בחשבון לפני יריית הפתיחה למרוץ הרגרסיה הוא גודל תכולת הספרינט. לא תמיד יהיה זה חכם בכל ספרינט לצאת להרפתקה של ריצה על כל תכולת הבדיקות. במידה והיו פיתוחים קטנים או מעטים, וכמו כן מעט באגים, יש לתת את הדעת מה תוקן ואיפה ובהתאם להחליט על התכולה. לעומת זאת ישנם מצבים של שחרורי גרסאות רחבות, שמכילות כמה ספרינטים, הרבה פיתוחים גדולים, הרבה תיקוני באגים, שכל אלה יותר דוחפים לתת את הדעת להגדיל כמה שיותר את תכולת הבדיקות.



ניסים אריאל

ראש צוות בדיקות בחברת X-trodes, מספר שנים בתחום הבדיקות, נשוי+6 וחובב היסטוריה ופילוסופיה



פקטור נוסף שאינו פחות חשוב, הוא אופי הפיתוח החדש שמוזג לתוך המוצר. לעיתים ישנם פיתוחים שנוגעים פחות בצד הפונקציונאלי של המוצר, אך נוגעים בשינויים תשתיתיים. למשל, עדכון גרסה של React או AngularJS Material – תשתיות שה Front End נשען עליהן, מעבר לעבודה עם קונטיינרים חדשים, מעבר Cloud חדש וכו'... כל אלה צריכים להילקח בחשבון על מנת להתאים

את אופי הבדיקות, שיבטיחו שהמעברים נעשו חלק וכראוי. למשל, בעדכון גרסה לתשתית UI כפי שהוזכר, קל יותר לבצע מעברי מסכים ולראות שלא השתבש דבר. לעומת זאת כשהקוד עובר לסביבה חדשה, הסברה נותנת שזה מצריך יותר מאשר לראות שה-UI תקין.

תכנון מקדים

לכאורה סעיף זה במאמר מיותר, מפני שכל האמור לעיל גורם לנו להבין עד כמה התכנון חשוב. אבל חשוב לי בכל זאת להדגיש, התכנון חשוב מאוד. מפני שהדבר הקל ביותר הוא לדרוש ביצוע של הרצת כל תכולת הבדיקות. מנהלים רבים סבורים שכך מקטינים סיכונים, כך סוגרים כל פינה אפשרית מפני שהכל ייבדק, אף גורם לא יוכל לבוא אל האחראי על הבדיקות בטענות כגון – למה דבר זה או אחר לא נבדק.

פה מונחת לדעתי טעות, שעלולה חלילה להיות גורם להרס רב לאיכות המוצר ולכל תהליך הבדיקות בכלל. ופה אני נכנס לשלב הבא במאמר.

"לא תמיד יהיה זה חכם בכל ספרינט לצאת להרפתקה של ריצה על כל תכולת הבדיקות. במידה והיו פיתוחים קטנים או מעטים, וכמו כן מעט באגים, יש לתת את הדעת מה תוקן ואיפה ובהתאם להחליט על התכולה"

מגבלות הכוח

צוות הבדיקות המסור והמחויב ביותר, במצב נתון לדוגמא של מוצר גדול מאוד, שכל שבועיים נכנס אחרי ספרינט לבדיקות רגרסיה של כל תכולת הבדיקות – יקלע למצב לא חיובי. אסביר.

מכיוון שאני מדבר במאמר על זה על בדיקות ידניות, הגורם האנושי חייב להיות בשיקול. מעמסה גדולה ורחבה שחוזרת על עצמה בתדירות גבוהה מאוד ובאיטראטיביות סיזיפית - מובילה בהכרח בסופו של דבר לשחיקה של צוות הבדיקות.

העניין שהבודקים ייגלו במשימה ילך וירד. תחושת המרמור עקב הצורך האנושי הבלתי ניתן לביטול בגיוון ועניין - תלך ותגדל. במצב קצה כמתואר לעיל, אין אפשרות להימלט מהתוצאה הזו.

אך נשאלת השאלה, הרי חובה עלינו להבטיח את איכות המוצר? נכון. אך גם חובה עלינו להבטיח את תחושתם הרעננה באופן סביר של הבודקים. וללא תחושה זו נקבל בדיקות שאפילו אם הן כתובות באופן איכותי, הכרח הוא שהן ייעשו שלא מתוך הקפדה יתירה, כלאחר יד ועם חשש גדול לתשומת לב נמוכה.

ולכן אני שב להדגיש את העניין החשוב ביותר של תכנון בדיקות הרגרסיה. יש לקחת בחשבון בתכנון זה את כל המשתנים: כמות הבודקים, מחויבותם, גודל המוצר, כמות הפיצ'רים החדשים, אופיים



החלום הוורוד

אם נוצר הרושם שאני מתנגד לבדיקות רגרסיה, אין כך הדבר. המסר שלי הוא, שכל תהליך שנעשה צריך להיות במידה נכונה ומשקל מתאים שיביא תועלת. בדיקות רגרסיה הן חשובות מאין כמותן ואין שום אפשרות להבטיח את איכות המוצר בלעדיו. אבל – תכנון רבותי, תכנון...

לטעמי ישנו כלי בארגז הכלים בגודל של ה-QA, שיכול להביא הרבה מאוד תועלת בנושא. כמוכן שאני מתכוון לאוטומציה. מערך אוטומציה מתוחזק ומתעדכן, יקטין באופן מאוד ניכר את הצורך במשימות סזיפיות, שוחקות ומתסכלות, ויפנה את הצוות לעסוק בכלים נוספים, למידה טובה יותר של המוצר וצרכיו, קידום אג'נדות, ובעיקר תחושה טובה ורעננה שבדיקות תוכנה הן דבר שיכול להיעשות בכיף, בשמחה ומתוך עניין גדול. מי לא ישמח ללמוד על כלים חדשים, כגון בדיקות עומסים, סניפרים, יותר API, יותר אוטומציה, זמן ללמוד את הפיצ'רים החדשים כראוי ולכתוב להם בדיקות איכותיות יותר... כל זה יכול להתאפשר כשיש יותר מרחב מחיה, אז העניין במוצר וחדוות הלמידה מקבלים מקום גדול בהרבה.

עד כמה אוטומציה צריכה להקיף, איך ומתי, גישות שונות ואסטרטגיות בנושא זה, הם כבר נושא למאמר אחר...

של הפיצ'רים החדשים, אופי האיטראציה של הרגרסיות בחברה, וכל גורם בעל משמעות, שיכול להשפיע על תמונת המצב של המשימה שלפנינו.

סבורני, שיש אפשרות כמעט בכל מצב שיש בו צוות מלא, להתאים את הרגרסיה באופן שתהיה מקיפה מספיק מחד ויעילה מאידך. כל שבירה קיצונית לצד אחד תהווה בשלב מסוים פגיעה באיכות המוצר. ויותר יתר כמובן בתכולת הבדיקות - אין צורך לבאר את קלקלתו, אך גם אצבע קלה על ההדק שמפילה על הצוות שיגרה כללית לא נעימה בעליל בעבודתם - תפגע בהכרח ברקמה האנושית וממילא באיכות של המוצר.

עוד אספקט חשוב שיש לציין, הוא קידום האג'נדה של האיכות בחברה. לכל ראש צוות או מנהל בדיקות יש אג'נדות שונות שהוא מעוניין להנחיל בארגון בכלל ובצוות בפרט. למשל, העלאת איכות כתיבת הבדיקות, על ידי טסט ריווי או הדרכה ישירה בנושא, בניית כלים שונים, כגון בדיקות API, חפיפת הצוות לאוטומציה וכו'... במצב של רגרסיות לא נגמרות, כפי שלצערי ראיתי בצוותים שונים, לא מאפשרת את הקידום של צוות הבדיקות אל עבר אופקים חדשים.

הצטערתי לגלות שישנם מקומות שמכנים את ה-QA – 'צוות רגרסיה'. כל ענייניו של החלק הזה של המאמר הוא להוציא מדעה זו, שמקטינה ומצמצמת את המקצוע שלנו, כי היא אינה מאפשרת התפתחות.

מגזין עולם הבדיקות



שתפו את הרעיונות והנסיון שלכם!!!

מגזין "עולם הבדיקות" מחפש כותבים וכותבות בעלי רעיונות, לשתף את הנסיון שלהם במגזין בגליונות עתידיים.

בנושאים הבאים:

- תכנון בדיקות
- ניהול בדיקות
- מאמרי מוטיבציה (איך הגעתי לתפקיד הנוכחי)
- אוטומציה
- SOFT SKILLS
- "איך מסבירים "מה זה בדיקות"?"
- טרנדים חדשים בעולם הבדיקות
- סיפורים מעניינים על בדיקות מחיי היום יום

לפרטים נוספים

לשלוח למייל:

ניצן
052-6718757

magazine@testingworld.co.il



איילת מלמד כהן

בעלת ניסיון של 19 שנה בעולם ה-QA, רוב השנים כמנהלת בכירה בסטארטאפים וחברות גדולות. בשנים האחרונות מאמנת לפיתוח מנהיגות עצמית וניהולית, שיפור מיומנויות ניהול, כניסה לתפקיד חדש או ההתמודדות עם אתגרי התפקיד הקיים. בנוסף לאימון, איילת מלווה סטארטאפים בכל הקשור לניהול איכות, בונה צוותים ואסטרטגיות QA מותאמות לארגון. מאמנת מוסמכת בינ"ל, בעלת תואר ראשון ושני במנהל עסקים וכלכלה מאוניברסיטת בר-אילן. אמא לשלושה וזוקפת לזכותם חלק גדול ממיומנויות הניהול שלה.



תמכו באחרים והכל במקביל לתפקיד הפורמלי שלהם בעבודה.

מצבי קיצון אלו חשפו עוד משהו חוץ מביצועים גבוהים. הם נתנו מקום לנטייה הטבעית שלנו, בזמנים האלו התמקדנו במה שאנחנו חושבים שהוא חשוב, והופענו מול אחרים, בצורה אותנטית יותר מדרך כלל.

באיזור הגאונות אנחנו מוצאים את המשמעות והחיבור שחסרים לנו בשוטף. שם אנו חווים הגשמה ומימוש עצמי.

מאוד סקרן אותי ליהות מהו איזור הגאונות שלי, ואיך אחרים יכולים לזהות את שלהם.

הנטייה הטבעית שלנו היא לחשוב שזה פשוט מה שאנו מצויינים בו וחזקים בו, אחרת לא היינו מגיעים להישגים גבוהים, אך זה לא בהכרח נכון.

במשך 12 שנה הקמתי צוותים והובלתי קבוצות QA. תפקיד שנהייתי ממנו, והתאים לאופי ולכישורים שלי. קיבלתי פידבקים מעולים והתקדמתי מארגון לארגון. הייתי באיזור המצויינות שלי. היה לי נוח שם, וגם לסביבה שלי. בו בזמן, הרגשתי תקופה ארוכה שנכון לשנות, רק לא ידעתי במדויק מה.

השאלות בספר הזה גרמו לי לעצור ולבדוק.

"המלחמה שנכפתה עלינו, ובאופן פרדוקסלי היא זרז להתפתחות. היא מאמנת אותנו בלהתמקד במה שבשליטתנו, וחושפת לנו בצורה מדויקת במה אנו מצויינים וגאונים בו."

הדרך לגילוי איזור הגאונות של כל אחד, היא להתחיל במודעות ומשמ להתחייבות לפעולה.

אלו השאלות, שחשוב לענות עליהן עם שלושה דגשים: בצורה אותנטית, כנה, כשאתם נטולי הסחות דעת, ובכתב. כשעולה לכם תשובה, תחשבו על כמה דוגמאות שמתקפות אותה.

- 1 מה החלק בעבודה שלך שלא מרגיש לך כמו עבודה (היית עושה אותו בכל מקרה, גם ללא תגמול כספי)?
- 2 מהו הדבר שאתם עושים, ומרגישים שזה הדבר הנכון, שאתם במקום הנכון ומה שאתם עושים ממש חשוב לכם?]
- 3 אם היה לי אומץ, הייתי הופך ל...

דוגמאות:

- אתם מבינים מהר דברים מורכבים - רואים את הבאגים עוד לפני שנכתב קוד
- מוצאים קשר בין חלקים לא קשורים, בדאטה, במערכת, בהתנהגות של אנשים.
- מחברים אנשים למשימות בקלות - יודעים להניע אנשים

לצאת מאיזור המצויינות

כמה פעמים ייעצו לכם, או שמעתם את עצמכם אומרים למישהו "לצאת מאיזור הנוחות?"

מניחה שהרבה. מה קורה כשאיזור הנוחות שלכם הוא גם משהו שאתם מצויינים בו? ואם אתם מצויינים באיזור הזה, אז למה לצאת? את השאלות האלו גם אני שאלתי את עצמי.

הרבה פעמים כשאנחנו ממש טובים במה שאנחנו עושים, מקבלים פידבקים טובים על הביצועים, אך רוצים לתפוס כיוון אחר או לתת מענה לשאיפה מקצועית חדשה - אוטומטית עולות שאלות, שבשיל מה? למה עכשיו? כמו גם חששות, של כולם בסביבה, בנוסף לשלך.

מצד אחד, איזה כיף לשבת בשיחת הערכה ולשמוע על הביצועים המצויינים שלכם? הסביבה מעודדת אותנו להמשיך במסלול שאנחנו נמצאים בו, דרך פידבק, יעדים חדשים בעבודה, תגמול כספי וזה אכן מחמיא. מצד שני, אם אתם לא במקום בו אתם חשים סיפוק ומשמעות ממה שאתם עושים, בפנים תתפתח שחיקה לעשייה ולחיבור לעצמכם.

אנחנו בימים של מלחמה. זאת שהגיעה מבחוץ, הפגישה כל אחד מאיתנו עם המלחמה הפנימית שלו. קשיים שכבר חווינו התעצמו, ואתגרים שלא היינו בוחרים לעצמינו, הופיעו.

"הפעולה באיזור הגאונות שלכם היא זאת שבלי הרבה מאמץ, אתם מייצרים השפעה גדולה"

בשבועיים הראשונים של הלחימה, פגשתי מנהלי QA ופיתוח שהיו צריכים לשחרר גרסה ללקוחות, בזמן שחלק גדול מהצוות וההנהלה בארגון גוייסו למילואים. היו מהנדסי איכות שמצאו עצמם לוקחים הרבה מעבר לתחום אחריותם, כאלו שקיבלו משימות של קולגות בצוות בנוסף לאלו שלהם והיו צריכים להסתדר עם כל אלו במקביל לאילוצים מהבית.

בן רגע, עברנו מדאגה לסטטוס הבאגים, לדאגה לקרובים או לסטטוס התרומות לחיילים.

מכולנו נדרש בימים אלו ניהול עצמי שהוא הרבה מעבר לשגרת יומנו. המצב הקיצוני, הכריח אותנו להתחבר יותר לעצמינו ולאחרים ונתן מקום להיות יותר אותנטיים, לבוא לידי ביטוי בדרך אחרת מדרך כלל, ולהתמקד במה שהכי חשוב לנו ושנאנחנו טובים בו, גם בלי ששמנו לב.

ההתפתחות שלנו קורית תוך כדי שאנו מתמודדים. ותשומת לב לאיך אנו מתפקדים בעיתות משבר, יכול לתת לנו רמז למושג חדש שכדאי שנכיר - "איזור הגאונות".

בספרו The Big Leap מתאר הפסיכולוג גיי הנדריקס 4 איזורים שבהם אנחנו פועלים ביום יום שלנו:

- 1 **איזור אי-מסוגלות:** הכולל משימות שאין לנו מושג איך לעשות אותן, כמו - לתקן מדיח.
- 2 **אזור מסוגלות:** הכולל משימות שאנו מסוגלים לעשות. כמו - לדבג את הקוד שלי, להכין STD.
- 3 **אזור מצויינות:** הכולל משימות שאנחנו טובים בהן ומקבלים פידבקים טובים על הביצוע שלהן מהסביבה, מההנהלה. למשל - רמת ההבנה במוצר, ניהול סיכונים איכות, כיסוי בדיקות מעולה, כישורי ניהול הצוות.
- 4 **אזור גאונות:** זהו האיזור שבו אנחנו מתפקדים בשיא שלנו, גאים בעשייה ובהישגים שלנו במסגרתה. פעולות שבהם אנו חווים משמעות עמוקה, חיבור לעשייה וחיבור לעצמנו.

מצבי קיצון, באופן פרדוקסלי, מציפים את התפקוד שלנו באיזור המצויינות - מי שלקחו על עצמם עוד אחריות הם אלו שיכלו לשאת אותה. מי שהחליף את המנהלת שלו היה זה שהיו לו כישורים לכך, מי שמצויין בעבודת ממשקים, עשה עבודות תיאום ואופרציה - בעבודה או בהתנדבות, במילואים או בבית. היו את אלו שהקשיבו,



STEP OUT Of Your COMFORT Zone

- מפרקים בעיות לחלקים קטנים - הופכים מטרה לתוכנית
 - מייצרים השראה מעצם היותכם
 - שואלים שאלות טובות, אתם אלו שמצליחים למקד כשכולם מתפזרים
 - מזקקים סיפור לשורה תחתונה מדויקת.
- הפעולה באיזור הגאונות שלכם היא זאת שבלי הרבה מאמץ, אתם מייצרים השפעה גדולה
- אם מה שזיהיתם באיזור הגאונות שלך כבר קורה, אתם בוודאי מרגישים את הפידבק הפנימי החיובי, שמהדהד לכם.
- כשאני עניתי על השאלות האלו, הבנתי שהעבודה האמיתית שלי, מעבר לכל מוצר או פרויקט, היא לעזור לאחרים להצליח. ולהיות באיזור הגאונות פרקטית אמר, להגביר את החלק הזה ביום יום - אז הלכתי להיות מנטורית, בהתנדבות, כחלק מהתפקיד הפורמלי בארגון או בכל דרך שבהמשך מצאתי.
- המלחמה שנכפתה עלינו, בהיבט הזה, ובאופן פרדוקסלי היא זר להתפתחות. היא מאמנת אותנו בלהתמקד במה שבשליטתנו, וחושפת לנו בצורה מדויקת במה אנו מצוינים וגאונים בו.
- קחו את הפעולה באיזור הגאונות האישי שלכם לפרקטיקה, גם אם זה רק לכמה דקות ביום. זהו אותו. יישמו אותו. בין אם בהתנדבות, ובין אם כחלק מהתפקיד - תיזמו פרויקט ביט אינטראקציות ופורומים שיאפשרו לכם לזרוח באיזור הגאונות האישי שלכם.

בחן את עצמך | טל פאר



הפתרון לשאלה

נתחיל עם יעד הלימוד (Learning Objective) ורמת הידע (Knowledge Level) הדרושים לנושא הזה. לידיעה כללית: את יעדי הלימוד ניתן למצוא בעמוד הפתיחה של כל פרק בתוכנית הלימוד (הסילבוס). לכל יעד לימוד מוגדרת רמת הידע הדרושה לו (K-level – Knowledge Level). המודל המייצג לרמות הידע שארגון ISTQB® משתמש בו הוא מודל הטקסונומיה של בלום (Bloom's taxonomy).

יעד הלימוד הרלוונטי לשאלה שלנו כאן הוא: FL-3.2.1 - "סכמו את הפעילויות של תהליך סקירת מוצר העבודה"; השאלה שלנו היא ברמה של K2. רמה זו אומרת שעל הנבחן להצביע על התשובה הנכונה מתוך הבנה או פרשנות של הנושא המסוים (או הנלמד במידה וניגש לבחינה לאחר הכשרה).

כאמור, נושא ביצוע סקירות לדרישות, סוגי הסקירות, ה"שחקנים" (בעלי העניין) השונים בסקירות ועוד, מוצגים בתוכנית הלימוד (הסילבוס) והוגדרו על ידי מומחי בדיקות לאורך שנים, בהתבסס על ניסיון רב בתכנון וביצוע בדיקות סטטיות. אחת מהדרכים היעילות לביצוע סוגי הסקירות הרשמיות היא שימוש ברשימות תיוג (checklist) - שעיקר מטרתן לוודא שהסוקר בודק את הרכיב הנסרק (לדוגמא: הדרישות) על בסיס רשימות התיוג, ומבצע את ההנחיות למציאת הפגמים בדרישות או במסמך הדרישות בהתאם לנקודות ו/או להנחיות לבדיקה/סקירה שרשומות ברשימות התיוג.

הבה נבחן את השאלה והתשובות:

ראשית, עיקר הדגש בשאלה הוא השימוש ברשימות תיוג במהלך הסקירה הרשמית - משמע, לא לפני תחילת היזום של הפגישה הרשמית של הסקירה, לא במהלך הישיבה ולא אחריה.

התשובה אינה נכונה. במהלך התכנון מחליטים האם נעשה שימוש ברשימות התיוג. הכנת הרשימות אינה חלק מהתכנון. כמו כן, הסוקרים אינם מעורבים בתכנון, ואינם אחראים ליצירת הרשימות. א

התשובה אינה נכונה. במהלך תקשורת בנוגע לסקירה, כל הפגמים הפוטנציאליים שזוהו בסקירה הפרטנית מדווחים. מילוי רשימות התיוג על ידי הסוקרים כבר מבוצע, אם בכלל, במהלך הסקירה הפרטנית. ב

התשובה אינה נכונה. במהלך ישיבת הסקירה הרשמית, הבודקים מודיעים על כל הפגמים הפוטנציאליים בתוצרהעבודה שהם כן זיהו במהלך הסקירה הפרטנית. דוחות ליקויים נוצרים רק במהלך פעילות התיקון והדיווח. ג

התשובה נכונה. ייזום הסקירה ("קיק-אוף") כרוך בהפצת תוצר העבודה וחומרים אחרים, ובין היתר גם הפצה של רשימות התיוג. ד

אם תרצו שאציג שאלת דוגמה בנושא מסוים או אם יש לכם שאלות אנא פנו אלי באימיל: aron.tsubery@itcb.org.il





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



"צ'ט מאפשר לתקשר בסביבה שבה לא נוח או לא אפשרי לנהל שיחה"

שיחת טלפון היא פולשנית. היא מאלצת את המקבל להפסיק לעשות מה שהוא עושה ולהתפתות לשיחה. לפעמים אנחנו לא בטוחים שמה שרצינו לברר מצדיק את ההפרעה, במיוחד כשאפשר לשאול בצ'ט. כיוון שעל מסרון לא חייבים לענות מיד, מרגיש יותר לגיטימי להשתמש במדיה הזאת בכל רגע נתון.

בשיחת טלפון, אם לא עונים לשיחה, שום דבר לא התקדם. לעומת זאת בצ'ט, גם אם המקבל מתעלם או לא זמין לקרוא את המסר כרגע, הרי שהשאלה או הבקשה עברה, כך שחצי מהתקשורת כבר התבצעה; התקדמנו קצת.

יעילות: שואלים שאלה ומקבלים תשובה. בלי הפטופטים של לפני ואחרי, דבר שהוא במיוחד משמעותי כשמדובר בחברים לעבודה שידועים בכך שהם מדברים יותר מיד...

כמובן שלא רק אני שמתי לב לעדיפות שמסר כתוב מקבל כיום. ישנם מחקרים בנושא, ונראה שלפחות בתחום של שירות לקוחות יש לצ'ט יתרונות רבים: ההתקשרות מהירה יותר (לא צריך לחייג, להמתין לשמיעת האפשרויות השונות, ולנסות להצליח להגיע למענה אנושי למרות מאמצי ההסוואה של האפשרות הזאת). צ'ט מאפשר לתעד את השיחה, מקל על קבלת הוראות ומייתר את הצורך להתמודד עם מבטא קשה להבנה של האדם שמטפל בכך. כתוצאה מכך, יש העדפה של לקוחות לקבל שירות בצ'ט. גם השימוש היום-יומי ברשתות החברתיות הפך את המדיה הכתובה למועדפת בהעברת מידע, דעות, רגשות ומה לא.

1 ראו סטטיסטיקות: <https://www.revechat.com/blog/live-chat-statistics> ומאמר בנושא <https://homebusinessmag.com/sales/customer-service/people-prefer-live-chat-support-over-phone-call>



תרימו טלפון!

בימים האחרונים התקיים דיון מאוד אקטיבי באחד מהערוצים של Teams. אחד מהמפתחים בפינלנד פרסם שורה ארוכה של שאלות לגבי הדרישות והעיצוב (design) של רכיב מסוים. הדיון התמשך על פני יומיים והיו בו לפחות שלושים הודעות: שאלות, תשובות, שאלות הבהרה על התשובות, שאלות שנבעו מחוסר הבנת הנקרא, ועוד ועוד. בשלב מסוים כתבתי בצ'ט: "הגיע זמן להרים טלפון". לקח עוד חצי שעה ו-15 הודעות עד שאחד מהמשתתפים הודיע (בצ'ט, כמובן) שהוא קבע שיחת ועידה. למרות זאת ההתכתבות נמשכה עוד כ-20 דקות גם אחרי ההודעה.

ב-1990, כשהתחלתי לעבוד באינטל, קיבלתי משרד. משרד... נו, נו... זו הייתה קובייה ב-open space בגודל של 2x3 מטר, עם מסוף דיגיטל VT220 (אותיות ירוקות על רקע שחור), מקלדת וטלפון שולחני. היו מחיצות בגובה 1.70 מ' בערך, ארון תיקיות, מגירות שולחן ואפילו לספרים וציוד פרטי.

חוץ מהמקלדת, הכל השתנה. היום, כשאני מואיל להראות את פני במשרד (אני בעיקר עובד מהבית), יש לי שולחן של מטר וחצי, באזור משותף של ששה עובדים. רוב הציוד האחר גם הוא איננו – והאמת שחלק ממנו כבר לא נחוץ (למי יש צורך היום בארון תיקיות?). גם הטלפון נעלם באיזשהו שלב, ועברנו לאפליקציית טלפון על המחשב.

השינויים היו לא רק פיזיים. עם הזמן, גם דרכי התקשורת שלנו השתנו.

בעבר, אם לא ידעתי משהו, היו לי שלוש אפשרויות:

- לקום מהכיסא וללכת למשרד של מי שאולי יודעים את התשובה ולשוחח איתם
- לשלוח אימייל
- לטלפן

ואילו היום? יש לי לפחות שש אפשרויות:

- לחפש בגוגל
- לשלוח אימייל
- לשלוח שאלה ב-channel של Teams (או Slack) או מה שאתם מעדיפים)
- לשלוח WhatsApp

וכמובן אפשר עדיין לקום מהכיסא או לטלפן אמממ... לא מדויק. אם פעם הייתי פשוט מחייג למישהו ומקווה שהם יענו, היום נהייה מין כלל לא כתוב שלא מטלפנים לפני ששואלים ב-chat אם אפשר לטלפן. ובמקביל, תופעת השרשור הארוך של דיונים דרך מיילים או צ'ט קורית לעיתים קרובות, גם כשברור ששיחה הייתה פותרת את העניין בצורה יותר טובה.

למה זה קורה?

הנה כמה מחשבות בנושא.

- צ'ט מאפשר לתקשר בסביבה שבה לא נוח או לא אפשרי לנהל שיחה. זה אומר שאנחנו זמינים לחברי צוות גם כשאנחנו בעצם לא זמינים (למשל: תוך כדי השתתפות בפגישת עבודה), וזה מתחבר טוב עם ההשפעה של הטכנולוגיה על הציפיה שלנו לקבל כל דבר – כולל תשובות לשאלות – באופן מיידי.
- הדור הצעיר פיתח מיומנויות צ'טוט מופלאות בניהול שיחות מתחת לשולחן בבית הספר או תוך כדי שמירות בצבא, וההרגל תורם לבחירת צ'ט כאמצעי מועדף לתקשורת.



איך זה קשור לבדיקות?

איפה כל זה פוגש אותנו? אני מוצא שהמקום הראשון שבו הרמת טלפון מייעלת מאוד את העבודה זה בטיפול בבאגים. בקבוצות בהם עבדתי תמיד הטפתי: "אם יש פינג-פונג של יותר משתי שאלות תשובות דרך מערכת ניהול הבאגים – תפסיקו לענות במערכת ותרמו טלפון". מקרים אחרים זה כשמבצעים סקירה של תוצרי עבודה. גם שם: אם רשמתם הערה, והיא נמחקה ללא הסבר מתאים; או שהתעלמו ממנה; או שמתפתח דיון... הכי יעיל זה לדבר. זה חוסך אי הבנות, מאפשר לוודא פרטים עד הסוף, והרבה פעמים השיחה מתפתחת לכיוונים ורעיונות חדשים לפתרון בעיות.

"אם יש פינג-פונג של יותר משתי שאלות ותשובות... תרימו טלפון"

יתרונות נוספים

מסתבר שיש לשיחה יתרונות נוספים, שאולי משמעותיים הרבה יותר.

במסגרת החיפוש שעשיתי לקראת כתיבת הטור, מצאתי **מאמר**² שמתאר סדרת ניסויים שבדקו מה המדיה המועדפת על משתתפי המחקר לצורך מטרה מסוימת ועד כמה הציפיות מהמדיה שנבחרה תאמו לתוצאות בפועל. המשתתפים במחקר התבקשו לחדש קשר עם אנשים שהיו איתם בקשר קרוב בעבר, אבל הקשר נותק. המשתתפים יכלו לבחור אם לחדש את הקשר בשיחת טלפון או באימייל וכן לציין מראש מה הציפיות שלהם: כמה הם ירגישו "מחויבים" אחרי האינטראקציה ועד כמה הם מצפים שהתקשורת תהיה מביכה. רוב המשתתפים ציפו לחיבור טוב יותר דרך שיחת טלפון, אבל הביעו חשש שהאינטראקציה בטלפון תהיה יותר מביכה מאשר באימייל. בהמשך הניסוי, קבעו לכל משתתף באיזה כלי לתקשר עם החבר הישן (טלפון או אימייל). לאחר ביצוע ההתקשרות, שאלו את המשתתפים שוב לדרג את רמת החיבור שחשו ועד כמה האינטראקציה הייתה מביכה. התוצאות הראו שכצפוי, שיחת טלפון נתנה תוצאות טובות יותר מבחינת הרגשת החיבור. מה שהפתיע היה שהציפייה לאינטראקציה מביכה יותר כאשר מתקשרים בטלפון לא התממשה והמדדים של "כמה זה היה מביך" היו פחות או יותר זהים בין טלפון לאימייל.

ידוע ממחקרים אחרים (בין השאר אלו של זוכי פרס נובל הישראליים, **כהנמו וטברסקי**) שאנשים נותנים משקל כבד יותר לתוצאות אפשריות שליליות. ואכן, למרות שהמשתתפים ציפו לחיבור טוב יותר בטלפון, רבים העדיפו לתקשר באימייל על מנת להימנע מהמבוכה הרבה יותר שלה ציפו. אחת המסקנות של החוקרים מהתוצאות (ומתוצאות של ניסויים נוספים שערכו) היא שאנשים עלולים לבחור במדיה לא אופטימלית ליצירת קשר, מתוך ציפיות לא נכונות לגבי ההשפעה של המדיה על איכות התוצאה.

המרחק בין המחקר הזה לתקשורת היום-יומית שלנו בענייני עבודה, שהיא ברובה בנושאים טכנולוגיים, נראית רחוקה. ואולי לא? מחקרים אחרים שצוטטו במאמר הצביעו על כך שאנשים תופשים את בני שיחם כאינטליגנטיים, רציונאליים וחכמים יותר אם הם **שומעים** אותם, מאשר כשהם קוראים טקסט שמכיל את אותם המילים. ההנחה היא שהרובד הנוסף ששיחה מאפשרת – כגון טון הדיבור וההפסקות – מסייע להעביר תוכן בצורה אפקטיבית יותר. תוצאות דומות התקבלו כאשר מועמדים לעבודה התבקשו לתת "נאום מעלית" (elevator pitch). מועמדים **שהשמיעו** את הנאום שלהם הוערכו גבוה יותר ממועמדים שמסרו את אותו נאום בדיוק, בכתב.

האמת שכולנו מודעים למגבלות שיש לצ'ט בצד הרגשי ובגלל זה נוהגים להוסיף אימוג'ים, במיוחד כשאנחנו חוששים שאולי לא יבינו את מה שכתבנו (למשל, אם כתבנו משהו ציני ואנחנו רוצים שיבינו שאנחנו מתבדחים).

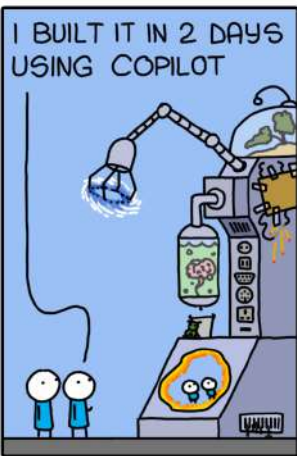
מסקנות ליום-יום

אני אמנם לא פסיכולוג או מומחה לתרבות ארגונית, אבל המסקנה שלי מהמחקרים האלה היא שיש חשיבות לשיחה בטלפון גם מעבר לעניין הספציפי של העברת מידע בצורה יעילה יותר. לדעתי המחקרים רומזים שיתכן ואנו טועים לעיתים כאשר אנו בוחרים לתקשר בצ'ט ולא בטלפון. כמו כן הם מצביעים על חשיבות אפשרית שיש לשימוש בטלפון בבניית צוות מגובש. אם שיחות טלפון מגבירות את הקשר בין אנשים וגורמות להם להעריך יותר אחד את השני, כדאי לעודד שימוש בטלפון, במיוחד בקבוצות שעובדות חלקית או מלא מהבית ושנאנשי הצוות מתראים In Real Life רק לעיתים.

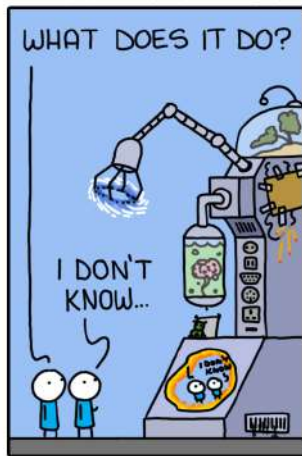
"יש חשיבות לשיחה בטלפון מעבר לעניין הספציפי של העברת מידע"

באופן אישי, ברגע שאני רואה ששיחה בצ'ט מתחילה להתארך, אני מציע לעבור לשיחה קולית. לכל מי שפונה אלי בצ'ט "האם אפשר להתקשר" אני אומר, כדבר ראשון בשיחה, שלהבא אין צורך לבקש רשות להתקשר. תתקשרו, אני אומר. מקסימום אני לא אוכל לענות. בנוסף, אני משתדל גם להדליק את המצלמה. אם זה משפר את התקשורת – למה לוותר על זה?

AI ASSISTANT



MONKEYUSER.COM



² https://www.researchgate.net/publication/344232654_It's_surprisingly_nice_to_hear_you_Misunderstanding_the_impact_of_communication_media_can_lead_to_suboptimal_choices_of_how_to_connect_with_others



משי מוטולה

בעלת תואר ראשון במדעי ההתנהגות ותואר ראשון בעבודה סוציאלית. ניסיון תעסוקתי כעובדת סוציאלית במחלקה לשירותים חברתיים מועצה מקומית בני עי"ש למשך עשר שנים ושנתיים של תיאום טיפול ושיקום במסגרת דיור למתמודדים עם תחלואה נפשית קשה. הוסמכת גם כמגשרת לאחר סיום קורס בדיקות ומבחן הסמכה של ISTQB® ומשמשת כבודקת פרילנס בתארי המונים נשואה ואמא להודיה.



בפרק יסודות בדיקות התוכנה מתוך הסילבוס של ISTQB® CTFL יש התייחסות לנושא פסיכולוגיה של הבדיקות. בתוך נושא זה מתחבאות סוגיות שונות הקשורות בטבע האנושי והשפעתו על תחום הבדיקות. מתן תשומת לב לפסיכולוגיה של הבדיקות עוזר לנו להתכונן למצבים רגישים שעלולים להופיע בסביבת העבודה ולהימנע מחלקם.

כעובדת סוציאלית ארצה להרחיב מזווית הראייה שלי על נושא זה. בפרט, ארצה להתייחס לנקודות הרגישות שבין בודקים למתכנתים בהקשר של דיווח על תקלות במוצר. אינני בדוגמה תיאורטית ברוח הפמיניזם לשם המחשה.

שרה היא מפתחת, רבקה היא בודקת. הן עובדות על תוכנה לעידוד העצמה אישית. התוכנה מאפשרת הקלטה של משפטי העצמה על ידי המשתמשת כדי לשמוע אותם מאוחר יותר. במהלך העבודה רבקה שמה לב שבכל יום בין השעות שתיים לארבע האפליקציה דוממת ואי אפשר לשמוע את ההקלטות, לאחר חקירה היא מגלה שהסיבה לכך היא שתהליך הגיבוי ששרה הוסיפה לאפליקציה לאחרונה רץ גם בשעות הצהריים ולא רק בין שתיים לארבע בלילה.

רבקה משוכנעת שהתקלה מחייבת דיווח מידי ברמת חומרה גבוהה. היא מתריעה על התקלה בדיווח כתוב להנהלה עוד לפני סיום יום העבודה שלה ויוצאת לביתה בתחושת סיפוק אדירה וחיוך רחב שמרוח לה על הפנים. כשהיא מגיעה למחרת למשרד, היא רואה לתדהמתה ששרה עושה לה פרצופים.

אז מה בעצם קרה כאן?

כשבאים להבין את ההתנהגות האנושית כדאי להיות מודעים להיבטים של התנהגות סמויה, הקיימים בנוסף להתנהגות הגלויה. הראשונה מתייחסת לתהליכים קוגניטיביים המתרחשים בנו. אנו מפנים קשב לגירויים בסביבה שלנו, קולטים אותם, מעבדים אותם ואז מגיבים אליהם.

גירוי הנקלט בחושים יצור גם עוררות בלתי נשלטת מסוגים שונים: פיזיולוגית ורגשית. בסוג הפיזיולוגי נמצא למשל שאנו מחייכים מבלתי התכוון או שהלב דופק באופן מואץ או שהזיעה תופרש יותר.

"אם לא יעשו משהו כדי ליישר את ההדורים ביניהן, יש סיכוי שכל סבב של תיקונים ובדיקות רק יגביר את הרגש השלילי האחת כלפי האחרת ובעקבותיו את שפת הגוף הקרירה, מה שיכניס אותן לסחרור של משוב שלילי"

"כדאי להיות מודעים להיבטים של התנהגות סמויה, הקיימים בנוסף להתנהגות הגלויה"

בעוררות מהסוג השני נמצא למשל רגשות של תסכול וכעס או לחלופין, שמחה והוקרת תודה. הפרשנות שניתן לעורריות האלו תשפיע גם על אופן התגובה שלנו, למשל "הדופק שלי עלה כי אני נלחץ בגללה וזה גורם לי לעבוד בפזיזות" או "הדופק שלי מהיר אבל זה בגלל שהיא עכשיו שיבחה אותי מול כולם אז כנראה שאני טובה במה שאני עושה ושווה לי להמשיך באותו אופן". ביחסים החברתיים המתקיימים ביחסי עבודה הגירויים יהיו למשל אמירות של מנהל או קולגה, הבעות הפנים, שפת הגוף, טון הדיבור, עוצמת הקול בה הדברים נאמרים, המהירות של אמירת המילים. אם התקשורת היא כתובה, ניתן להתייחס למלל כאל "בעל טון ועוצמת קול" משלו ובדמיון שלנו כבר "נראה" את הכותב עומד מולנו כועס, מאוכזב או לחלופין מסופק ושבע רצון. התגובות שלנו מושפעות ומשפיעות באופן מעגלי על היחסים המתמשכים במקום העבודה ויכולות להשפיע על איכות העבודה שלנו, כמובן גם על ההנאה מהעבודה ועל רמת השחיקה ממנה.

אנסה להמחיש את ההיבטים הפסיכולוגיים המתוארים כאן לפי הדוגמה שתוארה.

ניכנס לרגע לראשה של שרה. שרה מגיעה ליום העבודה שלה עם תחושה טובה בקשר לתהליך הפיתוח אותו היא אוהבת. שרה מבחינתה עשתה הכל כמו שצריך, היא עבדה במשך שבועיים על הפיתוח החדש, היא בחנה כמה חלופות ובחרה את הצורה המתאימה ביותר לכתוב את הקוד שלה, הציגה את התוכניות לכל הצוות, כתבה את הקוד, עברה סקירת קוד יחד עם ראש הצוות ווידאה שכל בדיקות היחידה עוברות. בסך הכל היא גאה בעבודה שעשתה ומרצה להתקדם למשימה הבאה ולהתחיל משהו חדש. כנראה ששרה ממסגרת בראשה חוויה נעימה של הצלחה ביחס לעבודתה, להשקעה הרבה שהשקיעה ולתוצרים שהשיגה. לתחושתה היא סיימה ובהצלחה. שרה אפילו הייתה מאחלת לעצמה לקבל איזו מילה טובה ומפרגנת על מאמציה והישגיה. אמנם, באופן שהפתיע אותה שרה מקבלת דווקא את הדיווח על תקלה בחומרה גבוהה שמעוררת בה

כלומר, אותה פעולה שמבחינת שרה היא עיכוב מרגיז, היא המטרה אליה חותרת רבקה. בנסיבות האלו טבעי שתגיע ההתנגשות. שרה ממש לא מפרגנת, ולא שותפה לשמחה שנמנעה מהלקוח תקלה, היא גם לא מבינה למה צריך להכריז על התקלה כ"חמורה" כשבסך הכל היה צריך תיקון של חצי שורה. שרה מתוסכלת שיש דופי בעבודתה, במיוחד אחרי כל המאמץ שהשקיעה. שרה משליכה את התסכול הזה על הקולגה רבקה. נדמה לה גם





סרטון או צילום מסך המשחזרים את התקלה עם הדגשה של השעון המובנה במסך.

במחשבה נוספת רבקה מרגישה שאולי הגזימה. יכול להיות שאפילו לא היה צורך בכל הדיפלומטיה הזו אילו רבקה הייתה פשוט קופצת לרגע לדבר עם שרה ומשתפת אותה מראש, כיוון שהתיקון קל, יכול להיות שאפילו לא היה צורך בפתיחת תקלה והתיקון היה נכנס בעקבות שיחה קצרה.

רבקה גם לומדת מניסיון זה שחשוב שכבודתך, היא תבקש להיות שותפה מלאה בכל שלבי הפיתוח. עוד לפני תחילת העבודה של שרה על הפרויקט, רבקה אמורה להצטרף לשיבת הצוות וביחד עם המנהלת חושבות, מה מטרות המוצר על פי הלקוח. בהתאם לכך רבקה צריכה להגדיר לעצמה שהמטרה שלה אינה למצוא באגים אלא לעזור לשרה ולשאר הצוות ליצור יחד את המוצר הכי טוב שהן יכולות ולקדם את העסק. מבחינה זו עדיף לה להיות חלק ממי שמסייעת למנוע מהטעויות להיכתב מלכתחילה. כך נחסך גם תסכולה של שרה שלא תיאלץ לתקן פרויקט שכבר הספיקה כביכול לסיים. רבקה יכולה להצטרף לשלב התכנון ולהגדיר יחד עם שרה ועם מנהלות המוצר מה הן ההתנהגויות שיהיה צריך לבדוק כך שכולן תהיינה בטוחות שהמטרות יושמו. היא יכולה גם להציע שינויים בעיצוב התוכנה שיהפכו אותה לפשוטה ככל הניתן או יוודאו שיהיה קל יותר לבדוק את המוצר. במצב כזה, גם אם יתברר בסוף שתקלה כלשהי הסתננה למוצר, ההסכמה שלהן על ההתנהגות הנדרשת מהמוצר תסייע להן בראיית התקלה והגדרתה ככזו, כלומר, שרה הופכת שותפה לרבקה באימוץ הלך החשיבה הביקורתי ואינה עוד חוששת ממנו, אולי אפילו שמחה על מציאת התקלה תהפוך משמחה שהיא כביכול שמחה לאיד רק של המוצאת, לשמחה משותפת, של כל העוסקות במלאכה. גם עבור המנהלת זו הקלה כי היא חולקת את האחריות בין כולן ומיישרת קו של הבנת הפרויקט עליו כולן עובדות. בהתאם לכך, כשרבקה תבדוק, יהיה לה קל יותר גם לבצע את הבדיקות וגם לגלות, כנראה, פחות תקלות.

"היבטים הקשורים בטבע האנושי יכולים לגרום לאתגרים במקצוע"

מחשבה נוספת שכדאי לרבקה ליזכר היא שטעויות הן דבר אנושי. אין ציפייה מאף אדם להיות חף מהן, במיוחד בתוך לוח הזמנים הצפוף שבפרוייקט תוכנה. חוויית לחץ שנובעת מלוח הזמנים הצפוף או מתחושת אחריות גדולה עלולים להגביר את הנטייה לטעות ולכן חשוב לזכור זאת ולחמול על הטועה, בין אם זה טעויות של שרה או בין אם שלנו. כאשר רבקה תפגש עם אותן תגובות ראשוניות של שרה ותזכיר לעצמה שהן לא נובעות מזדון, אז יהיה לה קל יותר לפעול כדי להנמיך את גובה הלהבות. אם תשכיל להבין את מקורן של אותן תגובות ולקבל אותן תוכל גם להבין ולקבל את הכאב שהן מביאות איתן ואז תצליח להתמודד יותר בקלות. בהתאם לכך, גם כלפי הטעויות האישיות שלה. אם רבקה למשל, בחרה במילים שיש בהן נימה מתנשאת, אזי, הכרה בטעות וחמלה עצמית חשובים מאוד כדי לאפשר יציאה ממעגליות שליליות פוטנציאלית ביחסים עם שרה.

אם רבקה עשתה כל שביכולתה והיחסים בכל זאת הסלימו היא תאשר לעצמה לבקש אוזן קשבת או סיוע באמצעות גישור מהגורם המתאים בהנהלה. זה נכון כמובן לכל מקום עבודה ובפרט בעבודה שבה היבט הביקורתיות קיים באופן מובנה ועלול להעכיר את האווירה.

אסכם ואומר שהיבטים הקשורים בטבע האנושי יכולים לגרום לאתגרים במקצוע הבדיקות אבל אם נקפיד להיות מודעים לתהליכים הקוגניטיביים המתרחשים בעת העבודה, נסכים לקבל בהבנה את התגובות הלא נשלטות והמכאיבות הקשורות בטבע האנושי, נאמץ פרקטיקות ממתנות ונבחר לגלות יותר רגישות וסבלנות, נגלה שניתן לצלוח את האתגרים.

שרבקה בוחרת להתבטא בצורה מאוד מתנשאת. היא לא יכולה להאשים את רבקה כי זה לא רציונלי, אבל על הפרצוף הנוקשה כלפיה כבר אין לה שליטה. היא מנסה לדבר איתה כמה שפחות, רק מה שחייבים, והיא לא מחבבת אותה, למרות שהיא יודעת שזה ילדותי. היא לא מצליחה אחרת.

השתיים ממשיכות בעבודה. שרה מתקנת ושולחת לבדיקה חוזרת. כבר פחות עובדת בהתלהבות, יותר בטינה.

רבקה גם מרגישה כבר פחות נעים, היא נמצאת בדיסוננס קוגניטיבי, היא לא רוצה לעצבן אף אחד אבל גם לא לטאטא בעיות מתחת לשטיח.

אם לא יעשו משהו כדי ליישר את ההדורים ביניהן, יש סיכוי שכל סבב של תיקונים ובדיקות רק יגביר את הרגש השלילי האחת כלפי האחרת ובעקבותיו את שפת הגוף הקרירה, מה שיכניס אותן לסחרור של משוב שלי. סביר גם שרבקה תרצה באופן לא מודע להפחית מעילות עבודתה כי עם הלקוח היא לא צריכה להתמודד יום יום והוא אינו מול הפרצוף שלה, לעומת הקולגה. מכאן יוצא שכל אחת מסיבותיה שלה עלולה לחוות שחיקה מהירה מהעבודה ובאופן טבעי קצב העבודה ויעילותה מצד שתיהן תלך ותפחת.

אז איך בכל זאת ניתן לפעול אחרת בתוך מגבלות הטבע האנושי? חלק מהתשובה מצריך גם בחינה של התרבות הארגונית במקום העבודה אך במאמר זה אתייחס רק להיבט האישי שבשליטתה של הבודקת בלי קשר לכך.

כיוון שמדובר בבעיה נפוצה, ישנם מספר עקרונות מקובלים בתעשייה שעוזרים לצמצם את החיכוך:

- ◆ מידע טכני ולא אישי.
- ◆ טון נייטרלי ולא מאשים.
- ◆ תקשורת ישירה ומטרה משותפת.
- ◆ מידע מקסימלי בעומס מינימלי.

נחזור לדוגמה, רבקה שואלת את עצמה מה היא יכולה לעשות אחרת. היא מסתכלת על דיווח התקלה ובודקת שלא חסרים בו פרטי הבהרה. היא מגלה שבעוד שהגדרת התקלה כחמורה הייתה החלטה נכונה מבחינה מקצועית, רבקה יכולה לראות שאין בתוכן שום הסבר לגבי הסיבה לחומרה. בעתיד היא תוכל להוסיף מידע כגון "שלושים אחוז מהמשתמשות הנוכחיות משתמשות באפליקציה בשעות האלה לפחות פעם בשבוע" שיסביר מדוע הבעיה חמורה למרות שהתיקון קל מאוד.

רבקה מזכרת בחיך שנמרח על פניה ועל הדופק המהיר כשחשבה לעצמה בגאווה "נו מה, היה לי ברור שיש פה איזו טעות שרק מחכה שאמצא אותה". במחשבה שנייה, זה פתאום נשמע לה כמחשבה יהירה וכשמחה לאיד. הרי יש פה קולגה שמשקיעה המון בדיוק בשביל לכתוב קוד ללא רבב וזה ממש לא מרגש (אותה בוודאי) למצוא בו תקלות. היא כמובן לא יכולה לפקוד על גופה להחליש את הדופק או למנוע מעצמה לחייך ולחלוטין מותר לה לשמוח על כך שעשתה עבודתה נאמנה, רק שהצלחה שלה היא בעצם סוג של כישלון בעיני שרה וטבעי שההתנגשות הזו תהיה לא נעימה למי שכביכול אחראי על התקלה.

רבקה תזכיר לעצמה שכדי למזער כמה שניתן את הביקורתיות האישית ששרה עלולה לחוש ממצאת תקלה בעבודתה, צריך להשתמש בדיווח במילים נייטרליות נטולות האשמה. הבחירה של רבקה לתאר את התקלה בתור "הגיבוי ששרה הוסיפה" הייתה חריגה לא נדרשת לתחום האישי והיה עדיף לומר "תהליך הגיבוי החדש". היא גם תמחק ניסוח שאומר "שכחת לבטל את הרצת הגיבוי בין שתיים לארבע בצהריים, היית אמורה לוודא שהגיבוי רץ רק בין שתיים לארבע בלילה". היא תעדיף מעתה ניסוח נייטרלי שאומר "הרצת הגיבוי היומי שמבטלת את יכולת ההקלטה על ידי המשתמש מתקיימת פעמיים ביממה בין השעות שתיים לארבע." לזה היא תצטרך



ניצן גולדנברג

מזה 8 שנים בתחום בדיקות התוכנה, תפקיד נוכחי מהנדס בדיקות בכיר בחברת SeatGeek, מנהל קבוצת ה-AB של ארגון ITCB[®] המוביל הראשי של קבוצת המיטאפ TestIL, מרצה בקורסים לבודקי תוכנה



עדכונים מערוץ הפודקסט הרישמי של TestIL Podcast מבית ITCB[®] את הערוץ מנהלים יחד נתנאל הרוש וקובי יונסי מוזמנים להאזין לכל הפרקים שיצאו ברבעון האחרון להנאתכם



פרק 7 – ראיון עם אסתר צבר - מקימת מכללת AQA

את AQA הקימה בשנת 2008 בעקבות היכרות עם התכנית הדנית של הזים טורקיל סונה. אסתר מסבירה את היתרונות, הקשיים, מה זה בכלל אוטיזם בתפקוד גבוה, ולמה דווקא יש התאמה מעולה לתחום בדיקות התוכנה והייטק בכללי. הביטוח הלאומי מכיר ב-AQA כתכנית שיקום ומפנה אליה מועמדים שעברו את האבחון התעסוקתי שהם עורכים ונמצאו מתאימים. כחלק מתהליך המועמדות לקורס בדיקות התוכנה, המועמדים עוברים סינון נוסף באמצעות תרגיל בית וסדרת ראיונות אישיים.

פרק 8 – ראיון עם יוני פלנר – בדיקות באמצעות ChatGPT

בואו לשמוע את יוני מספר על הכלי הכי חם כיום בשוק "ChatGPT", מה זה AI, מה זה "מודל שפה גדול" ואיך ניתן להשתמש ב-"ChatGPT" בשביל לתכנן את הבדיקות שלנו.

פרק 9 – תקלות ובחירת כלי בדיקה

בואו לשמוע את קובי יונסי מרצה מוביל בתחום הבדיקות ידבר על סקירה של סוגי תקלות אופייניות בפיתוח של תוכנה ועקרונות חשובים וכללים לדיווח יעיל של תקלות

פרק 10 – "מבוא לבדיקות נגישות" עם אלון פרידמן ויסברד

בואו להכיר ולשמוע את אלון פרידמן ויסברד, חבר בצוות הנגישות בחברת וויקס, מרצה במיטאפים של TestIL וכותב מאמרים במגזין "עולם הבדיקות" בפרק ראשון מתוך סט פרקים בנושא "בדיקות נגישות"

פרק 11 – "עושים רטרוספקטיבה לבדיקות" עם איילת מלמד

בואו להכיר ולשמוע את איילת מלמד-כהן משתפת דרך הסיפור האישי שלה, בכלים שמאפשרים לפתח את הקריירה, ללמוד מהצלחות ולהתגבר על הנטייה הטבעית שלנו, להתמקד בשלילי בפרק "עושים רטרוספקטיבה לקריירה".

פרק 12 – "בדיקות ממשק תכנות יישומים (API) עם גיל זילברפלד

בואו להכיר ולשמוע את גיל זילברפלד, מספר בפרק "בדיקות ממשק תכנות יישומים (API)" על הדרך שעבר מפיתוח, ולאחר מכן בדיקות ועד לרגע שהתחיל לייצג ובמהלך השנים התאהב בבדיקות API.

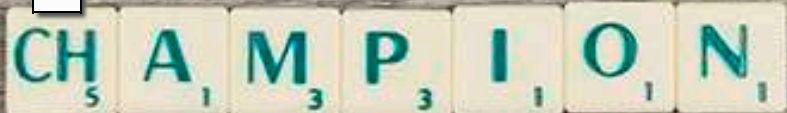
פרק 13 – ראיון עם מנהלת בדיקות נטליה רחמני

בואו להכיר את נטליה רחמני, מנהלת בדיקות בעברה ומנהלת מוצר בהווה מספרת על הדרך שעשתה בתחום הבדיקות ואיך היא עשתה את המעבר לעולם הפרודקט ואיך היא מצליחה לשלב את הנסיון שלה של בדיקות לעולם החדש שלה

מעוניינים להתראיין?
שלכם נושא שאתם מעוניינים לשקף?
בואו להתראיין לפודקסט
שלחנו לנו מייל ל
testilpodcast@gmail.com

אם גם אתם מעוניינים להשתתף בפודקסטים, אנא צרו עימנו קשר במייל: testilpodcast@gmail.com לערוץ הפודקסט שלנו

GAMBLING TESTING – תשבץ מושגים



Across

- 5. Testing to determine the performance of a software product.
- 10. The capability of the software product to provide the right or agreed results or effects with the needed degree of precision.
- 11. The capability of the software product to adhere to standards, conventions or regulations in laws and similar prescriptions.
- 13. A human action that produces an incorrect result.
- 15. Testing to determine if the game music and sound effects will engage the user in the game and enhance the game play.

Down

- 1. Part of quality management focused on providing confidence that quality requirements will be fulfilled.
- 2. A defect that was not detected by a testing activity that is supposed to find that defect.
- 3. A testing technique aiming to exploit security vulnerabilities (known or unknown) to gain unauthorized access.
- 4. The backup operational mode in which the functions of a system that becomes unavailable are assumed by a secondary system.
- 6. The degree to which two or more components or systems can exchange information and use the information that has been exchange
- 7. A procedure determining whether a person or a process is, in fact, who or what it is declared to be.
- 8. A measurement scale and the method used for measurement.
- 9. A part of a system that can be tested in isolation.
- 12. The degree to which a component or system is operational and accessible when required for use.
- 14. The status of a test result in which the actual result matches the expected result.





שי ביטון

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



שירה נוסבוים

הייטקיסטית ואמא במשרה מלאה, בדקות הבודדות שנשארות ביום בלוגרית אפיייה. 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.



משה מאמיה

בעל 17 שנות ניסיון כמהנדס, מתוכן מעל עשר שנות ניסיון ניהולי, מתמחה בפיתוח אוטומציה ובדיקות ביצועים. עובד מעל 5 שנים ב-HP כמנהל קבוצת QA ו-DevOps. חבר מייצע למועצת מנהלים של ISTQB® ומרצה בפקולטה להנדסת תוכנה במכללת .SCE



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.



אלכס קומנוב

בעל מספר שנים בתחום הבדיקות האוטומטיות. עובד כמפתח בדיקות ותשתיות אוטומציה בכיר בחברת SeatGeek נשוי+1 חובב נגינה על גיטרה וטיולים.



אלכסנדר זבולוקו

מפתח תשתיות אוטומציה בחברת SeatGeek. מתמחה ב-DevOps וטכנולוגיית ענן לעומק. מנצל את הידע והמיומנות לפיתרון בעיות ולהביא לחדירה מירבית של טכנולוגיות חדשות בסביבת התשתיות.

מחבר את המקצועיות עם התשוקה לטייל ברחבי העולם, מתרגש לחוות חוויות ותרבויות חדשות בזמן עבודה על מייזמים פרטיים.



עמית ורטהיימר

בודק תוכנה ב-Deep Instinct.



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



טל פאר

בעל ניסיון של יותר מ-25 שנים כבודק ומנהל בדיקות במגוון חברות וטכנולוגיות. כיום טל הוא יועץ בכיר ב-Grove Software Testing, חברת הדרכה מובילה בבריטניה וספקית של חומרי הדרכה. טל פעיל ב-ISTQB® והיה חבר בצוות המנהל של הארגון במשך 6 שנים. טל חבר בצוות המנהל של ITCB®.



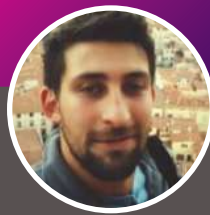
רחל ברוך

עובדת כבודקת תוכנה בכלל ביטוח. לאחר הפסקה של עשור מעולם התוכנה חזרה למקצוע הכי אהוב אליה. כשעובדים בעבודה שנהנים בה הזמן טס.



אסתר צבר

מהנדסת (.M.Sc.) בעלת 23 שנות ניסיון בפיתוח ובדיקות תוכנה, מתוכן 11 שנים בניהול QA בחברות ECI ו-BMC - ובנוסף חברה 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.



רוביק סביאניץ

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



דור רוזנברג

לפני כמה שנים החלטתי לעשות שינוי במקצוע שלי. לאחר מספר תחומי לימוד הכי הרבה התלהבת מלימודי בדיקות תוכנה. כרגע עובד בשרות לקוחות בישראל.