

עולם הבדיקות

www.testingworld.co.il

בדיקות בענן
קובי יונסי

מהו אתר תגובתי
(Responsive)
דניאל לוי

השתלבות של בודקים
בצוותי Agile משולבים
לנה הרמן שמוליאן

מדדי איכות KPI
בפרויקטי תוכנה
אליהו גלנצברג

איך לעקוף באג
בפרודקשן האנושי
אוהד יפרח



אנו מנסים לדמות תהליך עבודה כחלק מתהליך הלמידה. למשל, קורס מבוא לניהול פרויקט תוכנה הינו קורס המתבצע בקבוצות עבודה אגיליות אשר במשך סמסטר שלם עוברים הסטודנטים את כל תהליכי הפיתוח והיישום. בכל סבב פיתוח משמשים הסטודנטים בתפקידים שונים ונחשפים ומתרגלים מטלות ותיפקודים שונים בתהליכי הפיתוח והכל מתבצע בהדרכה ובלווי צמוד של המרצה והמתרגלים.

הקשר ושיתוף הפעולה עם התעשייה הוא המפתח להצלחתנו, אנו מחפשים שיתופי פעולה בכל תחומי הלימוד והפעילויות שלנו ושמחים מאוד לעבוד ביחד (בהמון אופנים) עם כל מי שמוכן להירתם למלאכה. חשובה לנו גם פתיחות המחשבה והיצירתיות, אנו מוצאים את עצמנו שוברים כל מודל פדגוגי ומחפשים כל הזמן רעיונות חדשים שיעזרו לנו להכשיר מהנדסים טובים יותר ויותר שמתאימים לסביבה המשתנה כה תכופות.

ישנם המון דוגמאות שונות אך אתמקד במאוצי פעילויות שמסיימים בדרך כלל פרקי לימודים והכשרה.

- **האקסאטון** - הנו מאוץ פעילויות ויצירה בקבוצה שנמשכות בדרך כלל 24 שעות רצופות. מעבר לערך הפדגוגי תרגולי זו חוויה ותהליך שלם שמוכיח בדרך כלל לקבוצה שביכולתה ליצור משהו חדש ושלם תוך כדי עבודת צוות מאומצת מסביב לשעון.
- **בדיקטון** - מאוץ או תחרות בדיקות שהוא תרגיל מסכם של קורס בדיקות שאנו מחייבים את מהנדסי התוכנה לקחת (מתבצע לפחות פעמיים בתהליך הכשרת מהנדס התוכנה). גם הוא תוצר של שיתוף פעולה עם התעשייה, חברה "תורמת" את המוצר שלה לבדיקה ועל הצוותים לבצע מבצע מסודר ומוגבל בזמן של בדיקתו. כאן בא לידי ביטוי לא רק היצירתיות הנדרשת לצורך תהליך הבדיקה אלא גם יכולת הדיווח הנכון והתרומה לאיכות המוצר. הצוותים נבחנים בראש ובראשונה באיכות הבאגים שמצאו ואופן הדיווח שלהם – כך שניתן יהיה להפיק תוצר איכותי שישפר את איכות המוצר שנבדק. ציון העבודה נקבע בעיקר ע"י הלקוח שבוחן את תוצאות הדיווח (כמות הבאגים שנמצאו, מה השפעתם על האיכות, איך ניתן לשחזר את הבדיקה ועוד).

*דוא"ל ליצירת שיתוף פעולה עם האקדמיה - SQC@ac.sce.ac.il

ברוכים הבאים לגיליון ה-14 של מגזין עולם הבדיקות

לאחרונה יש יותר פעילויות תחרותיות בבדיקות בישראל, משמח לראות שהקהילה יוזמת ומשתתפת.

ברבעון האחרון התקיימה תחרות הבדיקות הישראלית של ISTC בפעם השנייה, תוכלו לקרוא עוד על הפעילות והמשתתפים בקדם ובגמר התחרות והפרסים המכובדים להם זכו בכתבה המופיעה בגיליון זה.

תחרות נוספת קיימה בחודש מאי המחלקה להנדסת תוכנה ב-SCE המכללה להנדסה ע"ש סמי שמעון, האקדמיה בסימן מידע פתוח בעיר חכמה שכותרתו הייתה: Keeping the city Up@Data - שימוש במידע הציבורי לצורך שיפור החיים של האזרח. זה היה מבצע פתוח לציבור והשתתפו בו צוותים מכל הארץ. המפתח להצלחת מבצעים שכאלו הוא החיבור לתעשייה ובמקרה זה עריית באר שבע התגייסה ואפשרה גישה לכל המאגרים שלה. אלמנט נוסף הוא היכולת של המחלקה בגיוס מומחים (מנטורים) בעלי מומחיות שונה כדי לתמוך מקצועית בזמן האירוע. בהאקתון שהתקיים התנדבו 25 מומחים שהנחו ועזרו ל-20 קבוצות מתחרות. רעיונות רבים הוצעו ויושמו בצורה מבריקה ומאות הצופים שביקרו באירוע למדו שיתפו וחוו. לפרויקט הזוכה (מאשדוד) הייתה אפליקציה המאפשרת ומורה לאזרח היכן המקלט הקרוב בזמן אזעקה (כמובן עם מפה והנחיות גם למוגבלים) ובמקום השני זכה צוות מחיפה שעסק בהנגשה ושיתוף משאבים לציבורים שונים.

האקדמיה זה היה חלק מאופני הלימוד החדשים ושינוי תפיסת דרך הלימוד במסגרת לימודי הנדסת איכות, ביקשתי מדני אלמוג ממכללת סמי שמעון שיסביר לנו על הסיבות לשינוי דרך הלמידה והתפיסה החדשה:

"הבחננו שהפער בין מה ואיך שאנחנו מלמדים לבין מה שקורה עם הבוגרים הצעירים שלנו בתעשייה הוא מדהים, עברו הזמנים בהם מספיק היה לגרום לסטודנטים להתכונן לשיעור ולהרצות על נושא מסוים ולצפות שמכך ילמדו ויפנימו את הנדרש מהם. כיום מהנדס התוכנה (והבדיקות) עובד במציאות משתנה וצריך כל הזמן ללמוד תחומים חדשים. כמו כן העבודה המודרנית היא בצוותי עבודה וכישורי העבודה בצוות הנם מרכיב מרכזי בהצלחתם. התחושה המשותפת היא שמודל הלמידה הישן כבר אינו מתאים וצריך למצוא כלים ומנגנונים חדשים. וכך המוסד של SCE אומר לימודים ב-Project Oriented.



קריאה מהנה,

קוטליצקי איריס

תוכן העניינים

- 2 דבר העורכת
- 4 בחן את עצמך - שאלה - טל פאר
- 5 בדיקות בענן - פרק 1 - קובי יונסי
- 8 בדיקות מן העולם - תפקיד המנהל בעולם האגילי - אלון לינצקי
- 10... איך לעקוף באג בפרודקשן (האנושי) - אוהד יפרח
- 12 מדדי איכות KPI בפרויקט תוכנה - אליהו גלנצברג
- 14..... תחרות הבדיקות - אנה לוקובסקי
- 16..... אנציקלופדיה לבדיקות - שילוביות רציפה Continuous Integration - אייל זילברמן
- 17..... השתלבות של בודקים בצוותי Agile משולבים - לנה הרמן שמוליאן
- 19..... מחפש צרות - דם, יזע ודו"ח ביצועים - מיכאל שטאל
- 22..... פינת הטיפים - הקשב תחילה - ורק אז קבע עמדתך - קובי הלפרין
- 23..... נגיסה מ-TestIL - מפגשים לבודקי תוכנה - ניצן גולדנברג
- 25..... מהו אתר תגובתי (Responsive) - דניאל לוי
- 27..... בחן את עצמך - תשובה - טל פאר
- 28..... מה הסקרים אומרים - יאן ברון

מו"ל
Israeli Testing Certification Board
ITCB®

ניהול המגזין
יאן ברון, iMDsoft

ניהול התוכן
קובי הלפרין, Ceragon

עורכת
איריס קוטליצקי, Testing World

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

יצירת קשר
אימייל:
info.testingworld@gmail.com

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


www.testingworld.co.il


Israeli Testing
Certification Board

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 info.testingworld@gmail.com



טל פאר

בעל נסיון של מעל 20 שנים כבודק ומנהל בדיקות במגוון חברות וטכנולוגיות במודלי פיתוח שונים. טל עובד כיועץ עצמאי ומדריך בחברה Practical Testing שהקים, כשהוא מלווה ארגונים בהקמת צוותי בדיקות, הטמעת מתודולוגיית בדיקות, שיפור תהליכי בדיקה והדרכות בדיקות.

טל הינו חבר ב-ITCB® ומזכיר ארגון ISTQB®.



חלק גדול מפרויקט בדיקות תוכנה או מפעילות של בדיקות תוכנה, הוא הקמה וניהול של מכלול מרכיבי הבדיקה, הבוֹדָקָה או באנגלית Testware. לפי מילון המונחים של ISTQB®, שמופיע לצד הסילבוס, בודקה כוללת את כל התוצרים שמיוצרים בתהליך הבדיקות ומשמשים לתכנון, עיצוב, יישום, הערכה ודיווח על הבדיקות. למעשה כוללת הבודקה את מסמכי תכנון הבדיקות, מקרי הבדיקה, תהליכי הבדיקה הידניים והאוטומטיים, סביבות הבדיקות, דוחות הבדיקה וכו' (ההגדרה של ISTQB® מתבססת על תקן ISO 29119).

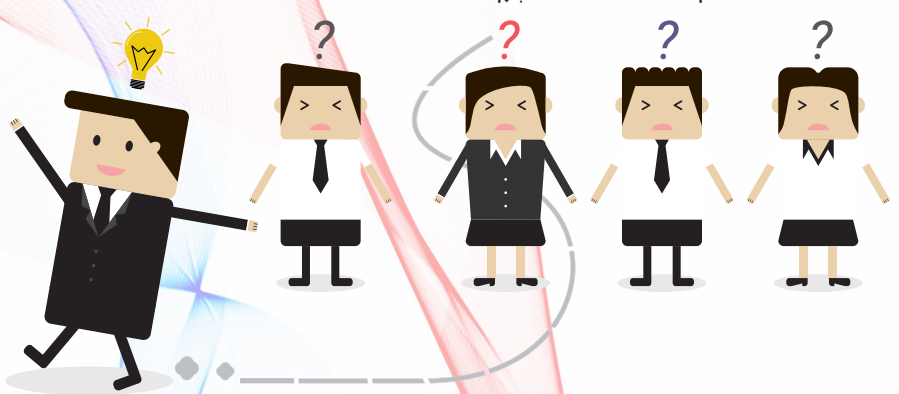
לאורך תהליך הבדיקות המפורט בסילבוס מפורטים השלבים הנוגעים להגדרה, הקמה וסגירה של מכלול מרכיבי הבדיקה.

ועל כך השאלה של היום:

אלו מהמשפטים הבאים נכון לגבי בודקה (Testware)?

הבודקה קשורה לגרסה מסוימת של מערכת, ויש ליצור אותה מחדש עם כל גרסה חדשה
יש לנהל את כל הבודקה במערכת ניהול תצורה (Configuration Management)
בסוף הפרויקט יש להעביר את כל הבודקה לארגון שאחראי על תחזוקה
המפתחים אחראים על התקנה נכונה של הבודקה

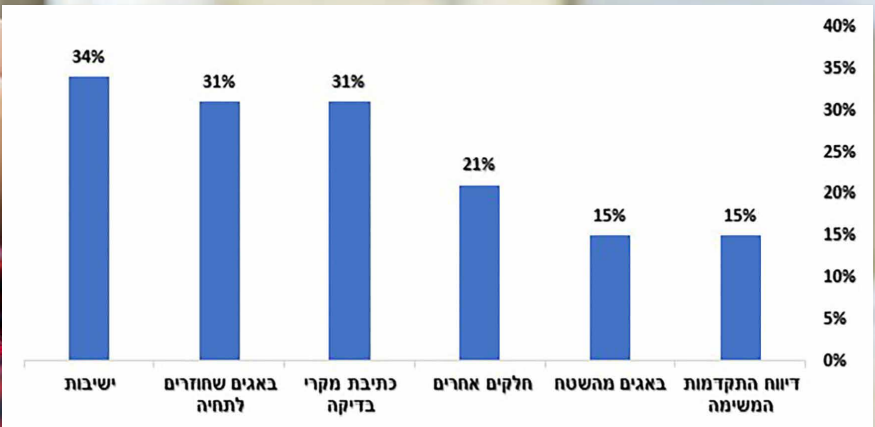
- I
- II
- III
- IV



- א. I, IV
- ב. II, III
- ג. I, III
- ד. II, IV

*לצפייה בתשובה המפורטת- דפדפו לעמוד 27

תוצאות ההצבעה מגיליון #13
מה האתג השנא עליך בבדיקות?





חלק ראשון



קובי יונסי

בעל תואר ראשון מאוניברסיטת בר-אילן
בתחום הלוגיסטיקה והכלכלה.

מוביל תחום לימודי בדיקות תוכנה
ביח' ללימודי חוץ של הטכניון.

מדריך ומרצה של בדיקות תוכנה
במגוון חברות מובילות, עמותות
חברתיות, מכללות ועוד.

יועץ עסקי ומלווה חברות בהקמה
וניהול של מערכי בודקים בארץ
ובחו"ל.



בעשור האחרון אנו עדים לשימוש הולך וגובר בטכנולוגיית הענן, או, במילים אחרות, שירותי מחשוב הניתנים לשימוש באמצעות מחשב מרוחק. כלומר, משתמש קצה מתחבר באמצעות האינטרנט או קו תקשורת ייעודי לשרתים מרוחקים ובכך נחסך ממנו הצורך להקים ולתחזק אותם. במקום לקנות תוכנה ולהתקינה על מחשב מקומי, ניתן להשתמש ביישום או במידע הנשמרים בחוות שרתים מרוחקות ולשלם (לעתים זה אפילו בחינם) רק לפי שימוש.

כיום, חברה או יזם שרוצים לצאת לדרך כבר לא חייבים לקנות שרתים יקרים כדי להקים את השירות או המוצר שבידם. זאת הסיבה שיש שיאמרו שמחשוב ענן הוא זרז לחדשנות כיון שמחיר ההימור הוא הרבה יותר נמוך מבעבר.

מה גרם להתפתחותה של הטכנולוגיה?

התשובה לשאלה הזאת מורכבת ממספר גורמים, חלקם קשורים לטכנולוגיות חדשות, לצרכים חדשים של לקוחות או ארגונים ולמגמות שמשפיעות על השוק:

- **ניתוח מאגרי מידע גדולים** - קיים צורך עולמי הולך וגדל להפקת ידע ערכי ממקורות מידע שונים בנפחים שונים ובפורמטים שונים. הדבר נעשה באמצעות טכנולוגיות כריית מידע מתקדמות ומערכות בינה מלאכותיות בזמן אמת. ניתוח מאגרי מידע גדולים דורש תשתיות שמסוגלות להתמודד עם הרבה מידע, יכולת שמתאימה לענן.
- **מידע חזותי עשיר** - הצורך הגובר לשיפור באיכות הצפייה בסרטים ברזולוציית ULTRA-HD ובאפשרות לראותם במסגרים שונים בו זמנית, מגביר את הצורך בניוד ושיתוף קבצי וידאו ענקיים (מעל 100GB לקובץ).
- **קטן, מהיר ונייד** - בשנים האחרונות אנו רואים מגמות גוברות להקטנת מוצרי צריכה תוך שימת דגש על עיצובם ושמירה על מרכיבי איכות (מהירות חישוב) וזמינות. לדוגמה, מכשירי סלולר, שעונים מתקדמים, משקפיים חכמים וטכנולוגיות לבישות נוספות. פלטפורמת הענן מאפשרת ליצרנים לתת מענה לדרישות השוק בצורה מיטבית ע"י הסטת מרכז הכובד החישובי ממכשיר הקצה לענן. הצפי הוא כי בשנים הקרובות כמעט כל מוצר אלקטרוני בו אנו משתמשים כיום יחובר לרשת כדי לאפשר לצרכנים יכולות שליטה, ניהול, ניתוח והפעלה מתקדמות ונוחות של מוצר הקצה.
- **היברידיות** - ארגונים רבים מעוניינים לפעול בסביבה היברידית, מצד אחד ענן פרטי המבטיח שליטה מלאה לארגון לצרכיו הפנימיים המיוחדים ובמקביל שימוש בענן ציבורי להחצנת שירותי הארגון ולגמישות תפעולית. כלומר, הצרכים הארגוניים מציבים דרישות חדשות ובמקביל לכך השימוש הגובר בעננים מאפשר לארגונים לפתח ולהרחיב את התשתיות כדי שיוכלו להשתמש בהן ביעילות. הענן לפיכך משמש כגשר בין פנים הארגון לסביבה העסקית בה הוא פועל ומבטיח רציפות עסקית ודינמיות המהווים תנאים הכרחיים להצלחת העסק.

כל המגמות בשוק שסקרתי מראות שהענן פה והוא משמעותי ולכן עלינו להכיר את הטכנולוגיה ואת מה שיש לה להציע.

מחשוב ענן מאפשר שירותים שונים אותם נהוג לחלק ל-3 קבוצות:

- **תשתית כשירות IaaS (Infrastructure as a Service)** – שכבת תשתית של משאבי אחסון ומחשוב המאפשרת למפתחים וארגוני IT להשתמש בתשתית טכנולוגית חיצונית לארגון במקום לרכוש אותה ולשלם עליה לפי היקף השימוש. דוגמאות: Amazon Web Services (AWS), Cisco Metapod, Microsoft Azure, Google Compute Engine (GCE), Joyent.

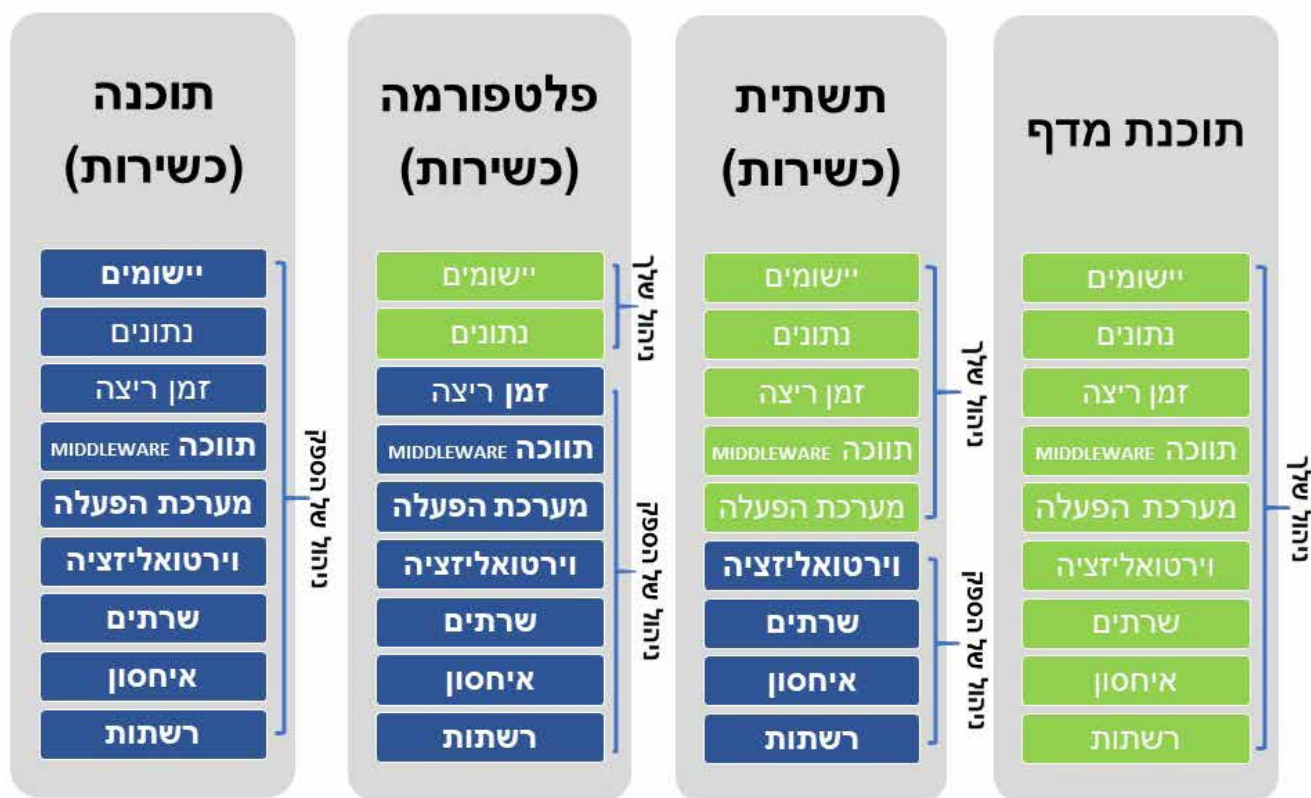
- **פלטפורמת כשירות PaaS (Platform as a Service)** – שכבה המעניקה למפתחים סביבת פיתוח כשירות המאפשרת להם לבנות אפליקציות ולפתח שירותים, כגון: שירותי גישה לנתונים, מסדי נתונים, שירותי חיוב, וכדומה, על גבי תשתית המחשוב בענן. דוגמאות: Apprenda, Platform.sh, Morpheus, IBM Cloud, Celigo, FLYnn.

- **תוכנת כשירות SaaS (Software as a Service)** – בשכבה זו ספק השירות מארח פתרונות אפליקטיביים, כגון: CRM, HR, ERP, עבודת הלקוח, ומאפשר לו להתחבר ולהשתמש ללא הצורך להתקין ולנהל את התוכנה ו/או לרכוש חומרה ייעודית. דוגמאות: Google Apps, Salesforce, Citrix GoToMeeting, Cisco WebEx.

בהיבט הבדיקות נציין שבדיקות ענן יכולות לכלול שלוש משמעויות:

- בדיקות הענן עצמו, כלומר, **בדיקות של יכולות הענן** - ניהול מכונות, מעקב אחרי תשלום, צריכה וכד'. בדיקות אלו מבוצעות (לרוב) ע"י ספקי ענן.
- **בדיקות בין עננים** - מטרתן לבדוק התנהגות של אפליקציות מבוססות ענן על פני שירותי ענן שונים.
- **בדיקות תוכנה בענן** - בדיקות תוכנה "רגילות" על אפליקציה שמשתמשת בתשתיות ענן.

“
תחום הבדיקות בענן
מביא עימו התמודדויות
חדשות בעולמם של
הבודקים
”



בדיקות תוכנה בענן

במאמר זה נתמקד בבדיקות תוכנה בענן, מה היתרונות שהם מספקים לנו?

1. יחס עלות תועלת גבוה יותר, תשלום רק על זמן שימוש בענן.
 2. אפשרות של בדיקות בין צוותים שמפוזרים גיאוגרפית.
 3. בדיקות מהירות יותר, יצירת סביבות עבודה בגמישות ובמהירות.
 4. הפחתת מאמצים של IT - התלות בתחזוקת שרתים משתנה לטובת ניהול משאבי ענן.
- התמונה הבאה מאפשרת לנו לבחון את סוגי הבדיקות הקיימים בבדיקות מוצרי ענן,

יש שיאמרו שמחשוב ענן הוא דבר לחדשנות כיוון שמחיר ההימור הוא הרבה יותר נמוך מבעבר



בחינה של התמונה מראה שלא קיים הבדל מהותי בין הבדיקות הפונקציונליות והלא פונקציונליות בסביבת הענן מאלה שאנו מכירים מכל טכנולוגיה אחרת.



ועדיין, יש מספר נושאים שכדאי להתמקד בהם:

בדיקות תאימות - כיוון שעיסוק מרכזי בענן הוא יצירת סביבות בדיקה שונות והתאמתן למערכות הפעלה מגוונות. (Multi-Tenancy Testing) - בדיקה שנועדה לוודא אבטחה מספקת ובקרת גישה לנתונים כאשר משתמשים מרובים משתמשים במופע יחיד, כלומר עושים שימוש באותו אובייקט בזמן הרצת תוכנית.

היבט חשוב ואחרון אליו אתייחס הוא נושא שרידות והתאוששות (Disaster recovery Testing) שמטרתו לוודא את התנהגות התוכנה במצבים בהם השירות קרס עקב עומס קיצוני, נפילת רשת, כשלים במערכת וכד'. חשוב להבין כמה מהר המערכת מתרעה על קריסה והאם היא מספקת חייוו על כמות המידע שהלך לאיבוד בזמן הקריסה.

לצד היתרונות שמניתי בשימוש בענן וסוגי הבדיקות שנתמקד בו, ישנם מספר אתגרים הניצבים על הפרק, וביניהם סוגיות באבטחת מידע. מחשוב ענן מציב מספר אתגרים בתחום אבטחת מידע עליהם יש לתת את הדעת בשלב הבדיקות. לדוגמה:

- כיצד ניתן לוודא שהתוכנה מאובטחת בתוך תשתית ענן של שירות צד שלישי?

- כיצד ניתן לוודא ענייני פרטיות בתשתית שמבוססת ענן?

- באילו מודלי-בדיקות וטכניקות-בדיקה ראוי להשתמש על מנת לוודא אבטחה מלאה לאורך כל הדרך בין האפליקציה, שירות הענן והלקוח?

בדיקות נסיגה (רגרסיה)

בתוכנה רגילה בדיקות נסיגה מתבצעות בדרך כלל לאחר ביצוע שינוי בקוד התוכנה או בסביבת ההרצה על מנת לוודא כי התוכנה ממשיכה לפעול באופן תקין. אולם, במחשוב ענן קיימת יכולת של ריבוי דיירים שמאפשרת לשתף משאבים בין התוכנות בסביבת הענן. כתוצאה מכך, ייתכן ששינוי בתוכנה מסוימת בענן ישפיע על תוכנה אחרת באותו הענן.

בנוסף, עבור מוצרי PAAS כדוגמה האחריות לביצוע עדכוני תוכנה מוטלת על ספק הענן, ולכן לא תמיד ידוע לנו מתי מתעדכנת גרסה של תוכנה תשתיתית בענן.

מפה נולד הקושי האמיתי להגדיר מתי בדיוק נכון להריץ בדיקות נסיגה.

בדיקות אינטגרציה

בחברות שנעשה בהן שימוש בענן, בודקי התוכנה נדרשים לתפעל קומבינציה של מס' יישומים ותוכנות כשירות בענן, חלקן בתוך הענן וחלקן מחוץ לענן. כשזה קורה, נאלצים הבודקים להסתמך על API של תוכנות אליהן תתמשק המערכת הנבדקת בנוסף על פרוטוקולי תקשורת. הסיטואציה הזאת מקשה על בדיקות ההתממשקות בין היישומים.

בניית תשתית בדיקות לפי צורך

האם לבנות את תשתית הבדיקות באופן שיטתי או באופן אוטומטי?

למרות שהתשתית תומכת בבנייה אוטומטית, אין לבודקים עדיין פתרון תומך החלטה באיזו דרך ניתן לבנות תשתית בדיקה באופן זול ויעיל יותר.

בדיקות ביצועים ועומס מדרגי (Scalability)

עבור תוכנות הממוקמות בענן יש להתייחס ביתר שאת להיבטים של עומס עולה ויורד, ביצועים במגמת שיפור ובמגמת נפילה ובמודלים שניתן להשתמש בכדי לתחקר אירועים אלו.

תחום הבדיקות בענן מביא עימו גם התמודדויות חדשות בעולמם של הבודקים ומכאן נולדות דרישות חדשות כמו:

- אימוץ מודלים חדשים וקריטריונים מעודכנים להצלחה, לדוגמה אימוץ שיטות חדשות לבדיקות אינטגרציה:

1. פרוטוקולי תקשורת בענן ו-API חוצי ענן.

2. API של התוכנה בענן והאינטגרציה מול מערכות ישנות (Legacy) מחוצה לו.

3. בדיקות אינטגרציה מקצה לקצה ליישום בין עננים שונים.

במקביל שינוי התפיסה ביחס לבדיקות ביצועים ובדיקות עומס מדרגי (Scalability), עומס עולה ויורד, בדיקות ביצועים בזמן שיפור ולקראת נפילה וכל זאת בכדי לבדוק עומסים דינמיים ותשתיות משתנות בענן.

- הגדרה מחודשת לרמת השירות החוזית (SLA) בין ספקים ללקוחות. חלק מהמגמה הזאת כולל את נושא האחריות על הבדיקות. למשל, עד היום ספקי התוכנה בענן התחייבו לאבטחה ברמת התשתית אך מגמה זאת משתנה ואותם ספקים יצטרכו לספק רמת איכות בדיקה (QoS) הכוללת פרטיות למשתמש, אבטחת טרנזקציות עסקיות ואבטחת מידע.

חשוב לציין שגם בתחום ה-TaaS (בדיקות כשירות) אנו עדים לכך שהבדיקות בענן יוצרות קושי בהגדרת איכות הבדיקות ובאימוץ מודל תשלום רלוונטי עבור בדיקות אלו.

- כלים ופלטפורמות חדשניות לבדיקות הן צורך חדש שמתעורר אצל בודקי תוכנות בענן.

בשל אופי הענן קיים למשל צורך אמיתי בבדיקות עומסים בזמן אמת. מכאן נולד הצורך בסימולטורים שמאפשרים לייצר כמות משתמשים ותנועת מידע על ידי הדמיה של משתמשים אמיתיים והנעת נתונים בענן.

במקביל, קיים ביקוש גם לבדיקות אינטגרציה בהתממשקות שבין תוכנות בענן ותוכנות שמחוצה לו.

- מתודות בדיקה משתנות ופתרונות חדשניים מתבקשים לסייע בהתמודדות עם הבדיקות בענן. הבולטת בשיטות אלה היא ולידציה מתמשכת ובדיקות רגרסיה לאורך תהליך הפיתוח (Continuous Validation & Regression Testing).

נושא נוסף הוא בדיקות אוטומטיות ליכולות פעולה הדדית בתחומי הענן (Cloud Interoperability), והמשמעות היא לוודא את יכולת האפליקציה לעבוד על פרוטוקולים שונים, לחצות עננים, להתממשק למערכות בענן וכד'.

לאחר הסקירה שעברתם ואחרי שהבנתם שהטכנולוגיה היא כאן להישאר, אזמין אתכם למאמר הבא בנושא בו אתייחס לכלים חדשים בתחום הבדיקות בענן, אערך השוואה ביניהם ואציע תוכנית בדיקה סטנדרטית למי שיוצא לדרך עם בדיקות בענן.

“
הענן פה והוא משמעותי
ולכן עלינו להכיר את
הטכנולוגיה ואת מה שיש
לה להציע
”



תפקיד המנהל בעולם האג'ילי

הקדמה

במקומות רבים בהם אני מסייע ומיינץ בעולם ובארץ, נשאלת השאלה מהו תפקיד המנהל בעולם האג'ילי, שכן על-פי התיאוריה, הצוות אמור להיות צוות בעל יכולות ניהול עצמאיות, חשיבה עצמאית, ויכולות קבלת החלטות עצמאית, כמו גם חוצה-פונקציונאליות (רב-תחומי ביכולות ובמיומנויות שלו).

משום כך, ארצה הפעם לסקור את תפקיד המנהל (ראש הצוות בעיקר) בעולם האג'ילי, להשוות אותו לתפקיד המקביל בעולם הניהול הרגיל, לעמוד על הנקודות החשובות בתפקוד צוותים. לסקור איך תפקיד המנהל בעולם אג'ילי משתנה, נוכח העברת חלק נכבד מתפקיד הניהול למצב של "ניהול עצמי", יוזמה ולקייח אחריות על ידי חברי הצוות עצמם. עוד אגע בשינוי הנדרש כאשר עוברים לעולם אג'ילי עם כמות גדולה של מהנדסים במערך פיתוח המוצר.

עולם אג'ילי – מה זה אומר?

בפורמטים אג'ילים שונים, אם זה Scrum או Kanban או פורמט אחר, מדברים על כך שהצוותים הינם חוצי-פונקציונאליות (Cross Functional), ומתנהלים עצמאית. המשמעות הינה - שמתקיים שיתוף מידע על בסיס קבוע בנושאים מקצועיים שונים, כך שכלל המהנדסים בקבוצה ידעו יותר מתחום התמחותם (חוצי-פונקציונאליות), הם חדורי יוזמה, יודעים להעלות בעיות מבעוד מועד, לנתח את התכולה והסיכונים, ליישם – ולהיעזר באחרים בקבוצה בעת הצורך, לנהל דיונים כשמתעוררות בעיות או סיבוכים עליהם לא חשב הצוות בזמן שתכנן יחד את הפיתוח של אותה איטרציה.

הדגש על ניהול עצמי של כל אחד מחברי הצוות ואח"כ גם ניהול עצמי של הצוות בינו לבין עצמו, היא נקודה מהותית להצלחת הצוות בביצוע המשימות שלקח על עצמו. כל אחד מחברי הצוות, תורם באופן קבוע לביצוע עבודתו ולהצלחת ביצוע העבודה של האחרים, ואמור להרגיש אחראי על עבודתו בדיוק באותו אופן שמרגיש אחראי על עבודת אחרים. כלומר, האחריות על איכות הינה של כל אחד מחברי הצוות ושל כולם יחד.

אכן עולם אוטופי... אבל מניסיוני אפשרי בהחלט לשאוף ולהגיע אליו.

תפקיד המנהל בעולם הניהול הקלאסי

נבחן את תפקיד המנהל בסביבות ניהול רגילות, כלומר מנהל שיש לו צוות עובדים, מהנדסים, שעליו לקדם את עבודת הצוות יחדיו בכדי להגיע למטרה משותפת. אחריות רבה נופלת על כתפי המנהל והוא למעשה נמדד על ידי מנהליו באופן הביצוע של הצוות – כיחידה אחת. עליו לדאוג להתפתחות המקצועית של חברי הצוות במידה רבה, לדאוג שיוזעם לבצע את כלל הפעילויות באופן היעיל והתועלת ביותר, כולל תכנון, אפיון, ביצוע, ניטור ומעקב – כל אחד בתחומו הוא – ולהשלים זאת על ידי התערבות מקצועית, פיקוח ובקרה, תוך כדי תרומה מניסיונו לקבלת החלטות – אותה הוא מבצע בעצמו.

אם נבחן את ההקבלה בין תפקידיו של מנהל הצוות בסביבת ניהול רגילה, לתפקיד חברי הצוות ותפקודם בעולם אג'ילי, נראה שהרבה מתפקיד המנהל הועברו לאחריות חברי הצוות.

המרכיבים להצלחת הפרוייקט והשינוי הנדרש לניהול בעולם האג'ילי

נשאלת השאלה – האם זה מספק? האם אפשרי לבצע האצלת תפקידים כזו ולהצליח במשימות באותה מידה (ואולי יותר)? מספר מרכיבים חשובים ביותר להצלחת כל פרוייקט של כל צוות הם עבודת צוות, תקשורת יעילה ותועלתית, שיתוף פעולה הדדי ותמיכה הדדית ומחויבות למטרה. מרכיבים אלו, אם "נבנה" אותם וניצוק אותם לחברי הצוות האג'ילי, ימנפו את עבודת הצוות ויהפכו אותו לגוף אחד הפועל למען מטרה משותפת, מרכזית, שכולם מרגישים שותפים למען הצלחתה. כלומר התשובה נעוצה בשאלה: האם נוכל להעביר לצוות ולחבריו את המרכיבים העיקריים והקריטיים הללו בצורה טובה. מובן שהאחריות על כך נמצאת אצל חברי הצוות עצמם, בבגרותם המקצועית, אבל רבות הדבר תלוי בתפקידו של ה-Scrum Master בעולם ה-Scrum. הוא החולייה המקשרת, המנטור שאמור להוביל את הצוות להיות כזה בנקודת זמן מסוימת, ופועל למען חברי הצוות, למען שיתופם, למען הגברת היכולת שלהם להיות בעלי יוזמה, ולקבלת החלטות יחד ולבד (בניגוד לצוות בעולם הניהול הרגיל כשיש בעיה פונים לראש הצוות לקבלת תשובות...).

בנוסף לנ"ל, יש לקחת בחשבון את ההתפתחות המקצועית של חברי הצוות והשיפור ביכולות המקצועיות שלהם לאורך זמן. נניח שיש בד"כ בודק אחד בצוות, איך הוא ידע לבדוק טוב יותר, מהיכן יקבל תובנות על שיטות חדשות, דרכים יעילות יותר לבדיקה, כלים חדשים שכדאי להשתמש בהם וכו'. כנ"ל לגבי המפתחים – איך הם ידעו לכתוב קוד טוב יותר, לבצע סקירות קוד טובות ויעילות יותר, ללמוד שיטות טובות יותר לבדיקות יחידה, לכלים בתחומם וכו'. לצורך כך, ראש הצוות שנראה שהינו מיותר, נכנס לפעולה. תפקיד ראש הצוות נשאר אך משתנה מהותית. הוא הופך להיות המנטור, התומך המקצועי שאמור לעזור לחברי הצוות לבצע את תפקידם טוב יותר – כל אחד לבדו וביחד. ראש הצוות הוא ה"דבק" המקצועי שאמור לגדל "גאוות יחידה" בקרב המהנדסים ששייכים כעת ל"קבוצת השווים מקצועית", אם אלו כל הבודקים (בצוותי האג'יל השונים בפיתוח המוצר) או אם אלו המפתחים (בשפת פיתוח מסוימת) וכו'. תפקידו השתנה מהותית, עדיין יש לו אנשים תחת אחריותו, אך כעת, הם פזורים על גבי צוותים שונים. כמובן, שראש צוות הבודקים, יתמוך שונה מראש צוות הפיתוח, ולכן לשניהם יש תפקיד מהותי בתמיכה מקצועית במהנדסים השונים.



אלון לינצקי

בעולם האג'ילי הרבה מתפקידי המנהל הועברו לאחריות חברי הצוות

מייסד ומנכ"ל QualityWize™. מומחה להנדסת תוכנה, הבטחת איכות ובדיקות במשך למעלה מ-30 שנה, יועץ ומלווה ארגוני לשיפור איכות פיתוח מוצרים, שיפור תהליכי בדיקות, הגדלת ההחזר על ההשקעה במו"פ, ויישום מתודולוגיה ושיטות עבודה טובות יותר המותאמות באופן ייחודי לצרכי הארגונים העיסקיים, הטכנולוגיים וההנדסיים. בעשר השנים האחרונות, משמש גם כמאמן אג'ילי, מיינץ ומלווה חברות בשינוי לפיתוח Agile, ויישום שיטות הנדסיות משופרות. מרצה בכיר בכנסים בינלאומיים, מייסד SIGIST ישראל ומייסד-שותף של ITCB. יו"ר השיווק של ISTQB, כותב-שותף של תוכנית ההסמכה ISTQB® Agile Tester, ושל תוכנית ההסמכה ISTQB® Advance Level Technical Agile Tester.



השינוי בניהול במעבר לאג'יל בקבוצות גדולות

אם אין לנו צוות אחד אלא שלושה או עשרים צוותים, האם אופן הניהול ישתנה?

אז הכל טוב ויפה ואנחנו יודעים איך צוות יכול להגיע למצב בו יוכל לקבל החלטות בעצמו, ואיך הדבר אפשרי כשחברי הצוות נרתמים למען המטרה הזו ומשתפים מידע באופן שוטף ותומכים אחד בשני. אנחנו יודעים איך תפקיד ראש הצוות משתנה, ואיך יהפוך למנטור ותומך, מלווה מקצועי עבור אנשיו. אבל מה יקרה אם אין לנו צוות אחד אלא שלושה או עשרים צוותים, האם המערך שיצרנו יתנהג באופן דומה, ואם לא, מה יחסר לו כדי להחזיר למצב הרצוי ומהי הקריטיות של תפקיד המנהל במערך החדש.

במערך גדול יותר, נכנסים אלמנטים של סנכרון ותיאום בין הצוותים השונים, הסתכלות גבוהה יותר על תוכנית העבודה, הסיכונים, חשיבה על תקשורת בין הצוותים, שיתוף מידע ברמה גבוהה יותר בין הצוותים, שיתוף עבודה ותיעדוף בין הצוותים ובניית תוכנית עבודה כזו שתתחשב בתלויות הפיתוח והבדיקה. בכדי להשיג את כל אלו, יש צורך ב...ניהול. תפקיד הניהול הקלאסי כולל את רוב האלמנטים הללו, ולכן חשוב להכניס אותם למערך התפעולי כדי להשיג את התוצאה המבוקשת – עמידה במטרת הפיתוח לאותה איטרציה. אך מכיוון שהצוותים עובדים במתכונת של שיתוף פנימי, עבודה פנימית עצמאית וכו' (כפי שהוזכר קודם), היישום של הכנסת האלמנטים הללו למערך יהיה שונה. לא בהכרח נידרש מנהל אחד שמסתכל על הכל, אלא חלוקת תחומי האחריות לאלמנטים הללו על ידי מספר אנשים, והשאת חלק מהם לחברי הצוות עצמם ול-Scrum Master. לדוגמה ניהול על ידי Scrum of Scrum נותן ביטוי ופתרון אחד אפשרי לאתגר שבפנינו. בפורמט זה נפגשים ה-Scrum Masters של הצוותים השונים באופן תדיר כדי לסנכרן את הפעילויות השונות, לשתף מידע, לתקשר ולהעביר את המסרים והמידע לחברי הצוותים השונים. פורמט אחר שרואים בשוק הינו עבודה עם Guilds, פורומים של קבוצות עניין/עבודה שמאוגדות על-פי תחום משותף – בדיקות, פיתוח JAVA, פיתוח MOBILE וכו'. לקבוצות אלו יש מנהל Guild שמרכז את העבודה, מייצר פגישות קבועות, מקדם עבודה משותפת בנושא, מקדם שיתוף מידע בנושא ותומך מקצועית במשתתפים באותו הפורום.

לסיכום

ניתן ליישם אלמנטים חשובים של תקשורת, עבודת צוות, העלאת מודעות צוותית, מחויבות אישית וצוותית ועוד, אצל חברי הצוות האג'ילי ולהביא אותו למצב של יוזם ומקבל החלטות עצמאית. תפקיד ראש הצוות (או ראשי הצוותים) במקרה זה הינו מנטור מקצועי יותר מאשר אחראי על ביצועי הצוות (כי הצוות אחראי על הביצועים של עצמו...). והעולם האג'ילי למעשה שינה את תפקיד המנהל, מיקצע אותו, "לקח" ממנו את האחריות בנושאים הפחות מקצועיים, ניהוליים. עדיין יש צורך במנהל אדמיניסטרטיבי, שידאג לכלל הנושאים הרכים, ויהווה גורם מקשר עם משאבי אנוש בחברה ויישם את ההחלטות בחברה לאותם אנשים שבאחריותו, כולל ביצוע הערכות שנתיות וכו'.

ישנן נקודות נוספות לא פחות חשובות שבהן לא נגעתי במאמר זה, כגון: איך כלל המערך מבצע תכנון מתאים לפעילות כלל הצוותים לאיטרציה, איך מתבצע פירוק העבודה לצוותים, איך מבצעים ניתוח סיכונים, ואיך שומרים על מעקב אחר העבודה עם תיקונים והתאמות תוך כדי, איך מתבצע דיווח נכון וכו'. ניסיתי לתת את הדעת על נושאים כלליים, אך כואבים, בראייה של תפקיד המנהל בעולם אג'ילי, אך יש לזכור שכל חברה יש לה את המצב העיסקי שלה, הבגרות המקצועית של אנשיה (והיכולות המקצועיות שלהם), התרבות שלה, ואופי העבודה שלה (למשל עבודה עם צוותים מעבר לים) – נושאים שמשפיעים רבות על אופי הפתרון הניהולי, ואופי יישום השתלבות הניהול במערך התפעולי/פיתוחי בכדי להשיג את המטרה.

העולם האג'ילי למעשה שינה את תפקיד המנהל, מיקצע אותו, "לקח" ממנו את האחריות בנושאים הפחות מקצועיים, ניהוליים.



מעוניין/ת ליצור תוכן מעניין ולשתף את קהל הבודקים בישראל? אנחנו מחפשים אותך!



אנו מחפשים אנשי בדיקות שירצו לתרום ולפרסם מאמר מקצועי במגזין "עולם הבדיקות".

גם אם אין לך ניסיון בכתיבה או אם את/ה מתקשה לבטא הרעיון שלך בכתב, אנחנו כאן לעזור!

בואו לקחת חלק בכתיבה לגיליונות המגזין

ליצירת קשר: info.testingworld@gmail.com



קצת על עצמי...

שמי **אוהד יפרח**, בן 41, נשוי לפזית ואב לדרור בן 4 ולליבי בת השנה. אני עובד בתפקיד ר"צ בדיקות בחברת **Gett**.
כובע נוסף שיש לי הוא קידום תעסוקה של אנשים עם מוגבלויות בעולם התעסוקה בכלל ובהייטק בפרט. אני חובב ספורט ושחקן לשעבר בליגת העל בכדורסל בכיסאות גלגלים.
אהה, כן, אני גם מרותק לכיסא גלגלים מגיל חצי שנה כתוצאה מחיסון, שהתוצאה הפיזית של זה שאני מרותק לכיסא גלגלים, ידי הימנית משותקת לחלוטין וכמו כן גם שתי רגליי, המונח הרפואי הוא שיתוק מוחין, מי שמכיר אותי יגיד שיש לי שריטה אבל אני לא זוקף את זה לשיתוק מוחין.



ארוכה הדרך אל האושר

מגיל צעיר הבנתי וקבלתי את מוגבלותי, חיפשתי דרך איך להצליח בזכות ולא בחסד, היה לי ברור שעולם המחשבים הוא בשבילי, ושם אני יכול לעשות את מה שאני הכי אוהב, וגם להצליח.

המחשב הראשון שנגעתי (והתאהבתי) היה **commodore 64**, אחרי זה כבר התקדמתי ל-XT turbo.

לאחר סיום התיכון, התלבטתי בין התנדבות או לימודים, אז בחרתי בלימודים. לפני שנים רבות, הייתה מכללה בשם סיוון, שם סיימתי לימודי תכנות בהצטיינות.

לאחר סיום לימודי בהצטיינות, הייתי בטוח שאני "הולך" (שלא לומר רץ) לכבוש את העולם, למצוא את מקום העבודה הראשון ולהצליח בגדול. התחלתי להתראיין במקומות רבים, ראיון ועוד ראיון ועוד אחד, והבנתי שאף אחד לא מסוגל להסתכל לי בגובה העיניים, (למרות שאני די נמוך בהתחשב שאני על כיסא גלגלים).

חשוב לציין, שמחר מאוד, עוד לפני שלב הראיונות, הבנתי שיש פער בין הלימודים להלכה למעשה, ושם אני רוצה להיות הכי טוב במה שאני עושה, מקומי הוא בעולם הבדיקות. הרצון להרוס ולחקור מה יכול לקרות אם נעשה דבר כזה או אחר, ולמצוא באגים שישפיעו על המשתמש פשוט גורמים לסיפוק שרק מי שבתחום יכול להבין את ההרגשה הזו, וזה גם מתאים ליכולת שלי להבחין בשינויים מאוד קטנים, אז פניתי לתחום הבדיקות.

אלו היו שנתיים ארוכות ומייאשות, שנתיים של המון ראיונות, שבמהלכן לא הצלחתי לגרום לאנשים לא לתייג ולקטלג אותי.

עד אשר חבר שמכיר חבר שמכיר את הדודה של... שסגרו את זה "בחלונות הגבוהים" שאני מתקבל ויהי מה.

היום הראשון של חיים חדשים

אז אחרי שנתיים של ראיונות, התקבלתי לעבוד כבודק תוכנה בחברת **ניס מערכות**, במוצר שנקרא Nice Vision!!

כולי נרגש ומחכה כבר שהיום הזה יגיע, האושר הזה שסוף סוף אוכל להוכיח, לעצמי כמובן ולאחרים, שהנכות, היא בעליל לא "גורם משפיע!!" ושאיני יכול להניב תוצר בדיוק כמו כולם, ואף יותר.

שילוב של אנשים עם מגבלות בחברה "רגילה" יוצר משהו טוב, משהו סובלני, האנשים מקבלים פרופורציה לחיים ולומדים שאין דבר כזה שאין דבר כזה, וזה משפיע על התוצר של העבודה באופן עקיף ולא מודע.

לא פעם פנו אלי ואמרו ששינתי לאנשים את צורת החשיבה, ואת הדרך בה הם חיים.

לאחר כמה חודשים פנו אליי וביקשו שאביא "עוד נכה", פה הבנתי שבאמת, הצלחתי לשנות משהו בתפיסה!

התקופה בניס הייתה מהטובות בחיי, צוות מגובש! מדהים! משהו בתולי שצורב לי ב-BIOS כמשהו טוב!

לאחר כשנתיים וחצי רציתי משהו אחר, להתקדם אולי לאוטומציה, לנהל, משהו חדש, אתגר חדש, וכמו רוב האנשים התחלתי לרחרח בחוץ.

הפעם הגעתי מעמדה אחרת, הנכות כבר לא היוותה עניין, צברתי ניסיון וידע, יש לי נכונות ללמוד ולהתקדם.

הראיונות נראו אחרת, מן הסתם, יכולתי לברור ולבחור ולא להסכים על כל מקום שרק יסכים לתת לי סיכוי.

אני חייב לציין שהסובבים אותי הרימו גבה, "מה עכשיו? עד שמצאת מקום טוב? אתה יכול להמשיך שם עוד הרבה שנים.. לך תדע מה יקרה..?" אבל אני בשלי, רציתי להתקדם!

למקום הבא שנחתתי קראו דמנטרה (יחד עם האיש והאגדה דורון בר שהיה הר"צ שלי בחלק מהתקופה) עד שהחליטו לקצץ חלק מהחברה ואני ביניהם, ובצדק, לא כ"כ התחברתי למוצר.

פתאום, מצאתי את עצמי בבית, מובטל, חותם בלשכה, מחפש עבודה בנרות, והמחשבות האם העזיבה מניס הייתה נכונה? למה נכנסתי לסרט הזה שוב? ישבתי (נו מה :) בבית 9 חודשים, ראיון ועוד ראיון, ועוד ראיון, והפעם זה אחרת, אני יודע על מה לענות ומה אני רוצה, ואחרי 9 חודשים מגיע לראיון ב-**ICQ**, מי לא מכיר את זה? "או...או...", התרגשות, הריאיון, כאילו תפור עליי, הרעב לעבוד, הרעב למצוא באגים (רק בודקים בנשמה יבינו את זה), הרעב לחזור למעגל העבודה עשו את שלהם, החיבור למוצר היה ברור מאליו.

ב-ICQ ביליתי כמעט 11 שנה, בית שני, חברים שאיתי עד היום, חוויות שילכו איתי עד הסוף. ביצעתי כמעט כל תפקיד אפשרי שקשור ל-QA, מבדיקות, דרך הובלה ועד איפיון בקטנה. במהלך השנים האלו נישאתי ליפה בנשים, 2 שולחנות מלאים בחברים מהעבודה. שלא נדבר על טיולים וערביי חברה, גרמו לי להרגיש שווה בן שווים, למרות ואולי גם בזכות המגבלה, בשום מקום שעבדתי עד עצם היום הזה לא הרגשתי "מיוחד", "שונה" או "חריג".

לאחר עשר+ שנים בחברה, הודיעו על סגירתה והעתקת הפעילות לרוסיה, חוויה לא נעימה, אבל תמיד מוציאים את הטוב מדברים לא נעימים, זה כנראה באופי שלי.

לאחר עשר+ שנים בחברה, הודיעו על סגירתה והעתקת הפעילות לרוסיה, חוויה לא נעימה, אבל תמיד מוציאים את הטוב מדברים לא נעימים, זה כנראה באופי שלי.

לאחר ברורים קטנים, מה, מי ולאן, חבר טוב, החליט לפנות את כיסאו, והמליץ עליי כמחליפו בחברת ניס, בפרויקט קסום ומאוד מעניין, עם אנשים איכותיים, והפעם, כמעט להרים הכל מאפס. אז חזרתי לאהבה הראשונה, ניס, פרפרים בבטן, כמו לאכול את האוכל הטעים מבית אמא אחרי טיול ארוך בחו"ל.

היו שם שנתיים קסומות, להקים משהו כמעט מאפס, לירוק דם כדי שזה יצליח. לאחר שנתיים, המוצר נסגר, מכה חזקה לבטן, כאילו לא חוויתי שנתיים לפני כן סגירת חברה. תחושת הכאב הזו לעולם לא תהיה מובנת למי שלא חווה את זה, וכמובן, הכל באהבה, כי ככה זה באהבה, חומס, עושים באהבה או לא עושים בכלל.

הרצון להרוס ולחקור מה יכול לקרות אם נעשה דבר כזה או אחר, ולמצוא באגים שישפיעו על המשתמש פשוט גורמים לסיפוק שרק מי שבתחום יכול להבין את ההרגשה הזו



כאן ב-Gett, התהליך קורה אחד לשלושה שבועות וזה מצריך סנכרון בין כל ראשי הצוותים ומנהלי הקבוצות, לדאוג שהכל קורה בתוך הזמן שתוכנן, לשקף סטטוס אמיתי ולהרים דגלים בזמן מטרופי! אין רגע דל ב-Gett, וזה עושה את כל העניין!

העבודה זה דבר חשוב אבל לא הכי חשוב

אז החיים הם לא רק Gett, ו-Gett כמקום עבודה גם מאפשר חיי משפחה וחיים בכלל.

חוץ מלמלא אחר ההוראות של האיש (פזית), אני גם אבא לדרור (4) ולליבי (1), לרוב אני מגיע לזמן ארוחת הערב ולמקלחות, ויש יום אבא אחד לשבוע, שבו אני מוציא את שניהם מהגן ומבלה את אחר הצהריים בכיף.

בנוסף לכל זה, אני מרצה כבר שנים, ההרצאה שלי מתמקדת בסיפור אישי, ודרכה אני מעביר מסר שאין דבר כזה שאין דבר כזה, ואפקט הפרפר הוא לקדם העסקת אנשים עם מוגבלויות בעולם התעסוקה בכלל והייטק בכלל.

בנוסף, עד שהבכור הגיע, שיחקתי כדורסל בליגת העל לכדורסל בכיסאות גלגלים מגיל 22 עד 37, עד שתליתי את הגלגלים:

בן דרורי - עובד ב-Gett שנתיים בתפקיד DevTester ומתוך זה שנה עובד תחת אוהד בצוות. " אוהד מבחינת מנהל הוא בן אדם שהמילה שלו חזקה מהכל, הוא יעשה הכל בשביל העובדים שתחתיו, והוא מניע תהליכים מצוין ולא עוצר לרגע!
אוהד מבחינת אדם הוא בן אדם חם ואוהב מצחיק ורגיש ומהכל הכל אופטימי מאוד!"

מה צופן העתיד...

איפה אני רואה עצמי בעוד 5 עד 10 שנים? שאלה קשה, מודה.

בהחלט אהיה מחובר לעולם הבדיקות וההרצאות, כי זה מה שאני עושה הכי טוב, אולי משלב את זה יחדיו?

אבל לפני כן, עולם האוטומציה והניהול חדש לי, יש לי עוד דרך ארוכה כדי להטביע את חותמי. יש כ"כ הרבה דברים שאפשר לקדם ולעשות כדי להקל על הבודקים והמפתחים, אז אני מאמין שיצטרפו לסבול אותי לעוד הרבה שנים בתחום.

חפתי מספיק אז לסיכום...

אז מה היה לנו פה? הייתי נער צעיר ושאפתן, נשארתי שאפתן + ר"צ בדיקות ב-Gett + אישה+2 ילדים+15 קילו בשקט.

לעולם לא ראיתי את מגבלתי, אך כולם ראו אותה, אך לאחר עשר דקות של שיחה הם הבינו שהפוסל במומו פוסל.

אני באמת חושב שהיום שוק העבודה בכלל והייטק בפרט הרבה יותר פתוח לקלוט ונכון לקלוט אנשים עם מוגבלות.

ואני מבין שיש פה הרבה שאלות בנושא, אבל תזכרו כלל אצבע, אם העובד לא עונה על הקריטריון, זה לא משנה אם יש או אין לו מגבלה. במילים אחרות, לא לתת הנחה בגלל ש...

אם תינתן ההנחה, זה יביא בבוא היום לתסכול לשני הצדדים.

אדוארד צינובוי - עובד ב-Gett 7 חודשים בתפקיד מנהל קבוצת פיתוח ומנהל הישיר של אוהד - "אוהד אדם מדהים ומעורר השראה, מבחינה מקצועית אוהד עושה עבודה מעולה ולוקח על עצמו בדיוק את אותם אחריות כמו כל ראש צוות בדיקות אחר בארגון".

עליות ומורדות

במהלך השנתיים הללו בניס, שני חברים טובים שעבדו איתי ב-ICQ (אחד שהיה המנהל הישיר שלי, והשני ר"צ מובייל), חפרו לי בראש, בוא ל-Gett, בוא ל-Gett.. ולי היה טוב כאמור.

לאחר שהפרויקט נסגר, כל הצוות התפזר לתוך ניס, ולמרות שתמיד עבדתי עם אנשים טובים, וזה חלק חשוב אצלי, הפעם לא התחברתי למוצר, והחלטתי להיכנע ולהצטרף ליחידת 669 בחברת Gett.

אז בסוף 2014 נחתתי ב-Gett, אותו ר"צ מובייל, ביקש שאביא איתי את הרוח שהייתה אז ב-ICQ, כי באמת היה שם משהו מיוחד.

הגעתי ל-Gett, למחלקת הפיתוח שכללה 40 אנשי סטארט-אפ (מפתחים ובודקים) מדהימים. שיטת העבודה פה מדהימה ודינמית, לחץ, real time, כל הזמן לומדים ומתקדמים, הצבת יעדים מטרופים, ויאללה אחרי לקרב.

לאחר שנה וחצי בתפקיד קודמתי להיות ר"צ. אם עד אותו רגע, האתגרים היו מטרופים, תחשבו לעצמכם, איזו הרגשת סיפוק, אחריות והאם הם בטוחים שבחרו באיש הנכון?

אחד מהדברים הכי כיפים בעבודה זו האפשרות לעבוד עם מחלקת האופרציה, בשטח, לבדוק את המערכת בזמן אמת, עם הנהגים, הנוסעים, הלחץ, לראות איך פיצ'ר כזה או אחר מוסיף או גורע מהשרות, במילה אחת, כיף!!

עד לא מזמן, כחלק מהאחריות שלי היה להיות regression owner, כלומר אחראי על תהליך, למי שפחות מכיר את תהליך הרגרסיה, זהו בעצם תהליך, שכל הפיתוח נעצר בנקודת זמן מסוימת, מתקנים את אותה הגרסה, ומבצעים על אותה הגרסה בדיקות שתוכננו מראש, לאותה הגרסה נכנסים אך ורק תיקונים לבאגים שנמצאו במהלך תהליך הרגרסיה.





אליהו גלנצברג

נשוי + 3

בעל ניסיון של 11 שנים בתחום הבדיקות. שירת ביחידת אופק בח"א כבודק תוכנה על מערכות GIS, עבד ב-Qualitest כבודק, ר"צ ומנהל תחום בדיקות בפרויקט בנק הפועלים.

כיום QA Manager ב-SAPIENS חטיבת ישראל

בצוות בדיקות יש עשרה בודקים, שכתבו ובדקו 100 תסריטי בדיקה. 30 תסריטים מתוכם נכשלו בסבב הראשון, והושלמו בשאר הסבבים, סה"כ נפתחו 40 תקלות.

אם היו נפתחים 30 תקלות במקום 40 שנפתחו בפועל, האם הגרסה הייתה יותר טובה? או שגם 30 תקלות זה הרבה מדי. האם קיימים אינדיקטורים לאיכות הפיתוח? אם כן, מהם?
האם קיימים אינדיקטורים לאיכות בדיקות המסירה, אם כן, מהם?
ומה לגבי איכות בדיקות הקבלה והיחס בזליגת תקלות בין סביבה לסביבה (QA,UAT,PROD)?
האם ניתן לכמת את התקלות מול כמות ההשקעה בגרסה?



אנחנו בודקי תוכנה אבל, מי בודק את הבודק?

יש לנו עולם של מתודולוגיות ומתודות עבודה לתצורות שונות של מחזור חיי פרויקט ולטכנולוגיות שונות, אנחנו חייבים שחלק מארגז הכלים יהיה בו רכיב של מדידה עוקבת אחר התנהלות של כל מחזור חיי פרויקט.

למעשה, אנחנו כמנהלים חייבים לבנות כלי מדידה מדויקים שנותנים ראייה עקבית בכדי למדוד את איכות העבודה בכל מחזור חיי הפרויקט.

במקרה שאין למנהל את הכלים הללו, נוצרות סיטואציות פחות טובות, והחלטות מתקבלות רק ע"פ תחושות בטן.

לדוגמה אספר על מקרה שקרה לי, השתתפתי בישיבת התייעלות, בשלב בחינת העלויות של העובדים ראיתי מנהל שמעמיד עובד לפיטורין, לבסוף עניין הקיצוצים נדחה.

לאחר שלושה חודשים נדרשנו שוב לקצץ ולדרג עובדים, אני לתומי חשבתי הרי בחרנו בפגישה הקודמת אז נמשיך את הפגישה הנוכחית מהמסקנות של הפגישה הקודמת, יש מצב לשינויים אבל הייתי בטוח שזה יהיה נקודה טובה להתחיל.

אך לתדהמתי, המנהל העמיד עובד אחר לפיטורין ואת העובד הקודם הוא הגדיר כמצטיין ושאינו יכול להסתדר בלעדיו. זה מעורר שאלה קשה, האם למנהלים יש את הכלים הנכונים לנהל ולעקוב אחרי ביצועי העובדים.

אנחנו צריכים לחזק את החוזקות של העובדים שלנו, ולמשב אותם. האם אנחנו יכולים לתת להם משוב ספציפי? לדבר איתם על משהו משמעותי? או לדבר על דברים ערטילאיים? הרי משיבות כאלו העובד יוצא ללא משוב משמעותי וקורקטי.

אלברט איינשטיין אמר "אי-שפיות זה לעשות אותו דבר פעם אחר פעם ולצפות לתוצאות שונות".

האויב של הטוב הוא המושלם

כל מנהל צריך לדעת ש"דבר שלא נמדד לא מנוהל" הרי אין ארגון או מנהל שחושבים שמדידה זה דבר שגוי, אם כך מהו האתגר במדידה?

יישום מדידה עקבית עלול לחשוף תצורות עבודה לקויות וישנם ארגונים שחשיפת הליקויים לא מתקבלת בברכה ובמיוחד כאשר יש מעקב אחר ההתנהלות.

עד כמה שברור לכולם שהדבר הנכון הוא לאמץ את המדידה ולבנות תוכנית לשיפורים, ניתן לראות שיש שיטה יחסית פשוטה להסרת המדידה מעל סדר היום. פשוט צריך לחפש מקרי קצה שמשפיעים על 0.01% מהמקרים ואז במקום לנהל דיון על המהות והפרשנות מתנהל ויכוח על העובדות. בכדי למנוע דיון על אמיתות העובדות יש להשתמש במדדים שנתונים בעלי ערכים מוחלטים.

למען התנעת מהלך של מדידת איכות, חייבים לרתום את כלל המנהלים, ולמרות שזהו אינטרס של ההנהלה הבכירה לא תמיד יש

את הגיבוי והרצון שלהם. ללא גיבוי המנהלים יהיה זה קשה עד חסר סיכוי להטמיע מהלך של מדידת איכות לארגון.

במידה והיכולת לרתום את המנהלים נתקלת בקשיים, יש להשתמש באסטרטגיה של בחירת מדד פשוט וברור בתקווה להטעמת תהליך מדידה משמעותי.

יישום

השלב הראשון והקריטי - עיבוד הנתונים

1. מהן תקלות סרק ומהן תקלות אמת?

2. האם מודדים ע"פ ציר זמן קלנדרי או בציר זמן של גרסה?

לא ניתן להציג שום נתון כאשר אין הסכמה על העובדות. וכשאינן הסכמה, תצא הרווח מהמפרשים של המהלך למדידה ושיפור.

ולכן - הדבר החשוב ביותר ביישום מדדי איכות - יצירת הסכמה הן על המהות והן על העובדות.

השלב השני - בחירת מספר מועט של מדדים פשוטים, ברורים ומוחלטים, המקיפים את כל תהליך מחזור חיי הפרויקט.

השלב השלישי - על כל מדד יש ליצור מטרות ויעדים מוסכמים

השלב הרביעי - הטמעה של מנגנון פשוט לבניית ושימוש במדדים לתכנון ולבקרה.

עקרונות הגישה

- בקרה אחר יישום אסטרטגיה- הגדרה ברורה של הפעילות והנעה לעמידה ביעדים,
- ניהול יעיל ובקרה אחר הפעילות השוטפת- מדידה שוטפת ובקרה אחר הנושאים החשובים ביותר לניהול הפעילות של הארגון.

יתרונות הגישה

- מבטיחה את יישום האסטרטגיה בכל שלבי הפרויקט
- מאפשרת שליטה ובקרה על התהליך האסטרטגי, עד לרמת העובד הבודד
- מאפשרת איתור כשלים במערכת עד לרמת שורש הבעיה והמנהל האחראי
- בנויה על מדידה כמותית של תוצאות, אשר מאפשרת מתן תמונה ברורה על ההתקדמות, ומשמשת כתמריץ לעובדים ולמנהלים





יישום ומדידה בפועל

מדד ראשון - איכות הפיתוח והבדיקות

מדד השני - תקלות סרק

מדד שלישי - איכות התקשורת בין צוותים

מדד רביעי - יעילות הפיתוח

מדד חמישי - איכות הפיתוח

המדד הראשון - איכות הפיתוח והבדיקות

החלק הפשוט במדד זה, הוא למדוד את בדיקות מסירה וקבלה - מכיוון שצוותים אלו מבצעים בדיקות רק על פיתוחים חדשים או גרסיה ולכן קל מאוד לדעת את ריכוזי התקלות.

החלק המורכב במדד, זה התקלות שנפתחות מסביבת production-ה.

המורכבות נובעת מכך שיש להוכיח שכל תקלה שנפתחה מצוות התמיכה, מדובר בזליגת תקלות מהגרסה האחרונה.

בניית המדד בפועל - סכימת כל התקלות אמת שנפתחו בגרסה, חלקי מספר ימי הפיתוח. כך ניתן לדעת מהו היחס בין מספר התקלות למשך הפיתוח וכן את כמות התקלות שנמצאו בבדיקות המסירה ביחס לבדיקות הקבלה. במצב זה, מתגלה חסר במדידה של תקלות שזולגו מ-Prod.

יש שתי אפשרויות לפתור חסר זה:

1. לפצל את המדד

כאשר מדד א' בודק את יחס התקלות בין צוות בדיקות המסירה לבין צוות בדיקות הקבלה של כל גרסה.

מדד ב' מודדים על פני תקופה ארוכה של שנה את כל התקלות שנפתחו מ-Prod ואז מודדים איכות לא ברמת גרסה ספציפית אלא ברמה של תרבות עבודה ארגונית.

היתרון של מדד ב' שהטעון כי התקלה לא נובעת מהגרסה האחרונה, אינו רלוונטי כי אנחנו בוחנים מגמה על פני ציר זמן ארוך. החיסרון הוא שאין בחינה של זליגת תקלות ביחס להשקעה בגרסה.

2. לקבע את נתוני ה-Prod לבדיקות

אפשרות א' לקבוע תקופת זמן מהתקנת הגרסה, שבו כל תקלה שנפתחת היא בעקבות הגרסה.

היתרון: פשטות, קביעת התקלות.

החיסרון: לקבוע את הפרק הזמן הנכון שבו כל התקלות שנפתחו הם בעקבות הגרסה.

אפשרות ב' לתחקר כל תקלה מאיזה גרסת פיתוח היא נובעת.

היתרון: קביעה מוחלטת מקור התקלה

החיסרון: זמן תיחקור

לסיכום: מדד זה בוחן את איכות הבדיקות והפיתוח ויש מס' פתרונות איך למדוד גם תקלות מ-Prod.

המדד השני - תקלות סרק

אמר לי פעם מנהל "פוטנציאל הנזק גדול ביותר זה עובד טיפש שגם חרוץ".

מדד זה הינו מדד אובייקטיבי, לראות איכות התקלות שנפתחות ע"



KPI



הצוותים עם יכולת רזולוציה עד לרמת העובד.

כאשר עובד פותח תקלות שהן סרק הוא חוטא פעמיים, הוא השחית את זמנו שלו ושל הפיתוח.

בניית המדד בפועל: לחלק את סך תקלות האמת חלקי סך כל התקלות שנפתחו פר כל צוותים (מסירה, קבלה ותמיכה).

הקושי ביצירת המדד: אלו תקלות יש להגדיר כתקלות ואלו הן סרק.

לדוגמה: תקלה שבוצע תיקון זה תקלת אמת, תקלה שנוגדת את האפיון או תפעול לקוי זה תקלת סרק.

הקושי נמצא למשל, בתקלה שגורמת לשינוי אפיון ואז מבחינת הבדיקות זה תקלה ומבחינת הפיתוח הם פיתחו לפי האפיון.

לסיכום: זהו מדד מרכזי, שניתן לראות נתונים עד לרמת העובד.

המדד השלישי - איכות התקשורת בין צוותים

בודק פותח תקלה ואז ישנם מקרים בהם מתחיל פיג פונג בין הבדיקות לפיתוח ולבסוף יש מס' אופציות נסגר עם תיקון /ללא תיקון/ תיקון נוסף... מכיוון שישנם מספר מקרים ובכל מקרה הרשלנות היא על גורם אחר.

ובכדי להציג תמונת אמת, יש לפצל את המדד לכל המקרים, רק שזה נהפך למדד ארוך וטרחני.

הפתרון: הרי אם בסופו של דבר יש מטרה אחת שהתקשורת בין הגורמים השונים בפרויקט יהיו תכליתיים וברורים ובלי ניהול התכתבויות בלתי פוסקות. והרי גם במקרה שהתקלה עברה פיג פונג אין סופי ולבסוף נדחתה, אז נכון שהבודק כילה את זמנו של הפיתוח לריק - הפיתוח הפסיד זמן יקר על תקשורת לקויה, האם הפיתוח לא היה יכול לגדוע את הפיג פונג? ברור שכן ולכן אנחנו רוצים למדוד את רמת התקשורת בין הצוותים בלי קשר לאיכות התקלה.

בניית המדד בפועל: יש לסכום את כמות המעברים בין הצוותים השונים, ללא קשר לאיכות התקלה.

לסיכום: את איכות התקלות אני כבר מודד במדד השני, במדד זה אני מודד את איכות התקשורת בין הצוותים.

המדד הרביעי - יעילות הפיתוח

ישנם מקרים בהם נפתחה תקלה ואז תוקנה ומתברר שהיא לא תוקנה טוב, ואז נדרש תיקון נוסף.

בניית המדד בפועל: יש לספור את כמות התיקונים שבוצעו בכרטיס, כאשר בד"כ מדובר במספר ממש נמוך של תקלות שיש בהם יותר מ-2 או 3 תיקונים, לכן אין טעם לחלק את זה בסך הכרטיסים שתוקנו, ויש לטפל נקודתי בכל תקלה שלא תוקנה בפעם הראשונה.

לסיכום: מדד זה, בוחן את איכות הפיתוח, יש לציין שישנם מקרים בהם מנצלים את פלטפורמת התקלות לעדכוני אפיון.

תהליך מדידה הינו תהליך חובה וקריטי שמנהל ה-AQ נדרש להוביל, והמנהל צריך להוביל את מהלך המדידה בצורה מוסכמת ומשכלת על מנת להוביל את כל הפרויקט למצוינות.



מזל טוב לאלופי הבדיקות של ישראל

פרסים יקרי ערך הוענקו לזוכים בגמר

מקום I - טיסה, אירוח והשתתפות בכנס [Agile Testing Days 2018](#) שיתקיים בפוטסדאם, גרמניה בנובמבר 2018.

מקום II - קורס מקצועי (עד 40 שעות) וטאבלט בחסות מכלל ג'ון ברייס

מקום III קורס מקצועי (עד 40 שעות) בחסות מכללת ג'ון ברייס

העולים לגמר זכו בכרטיס חנים

לכנס [QAWeek 2018](#)

זו השנה השנייה לתחרות הבדיקות הישראלית ISTC שהתקיימה בזכות יוזמתה של אנה לוקובסקי ובתמיכתם של ITCB - העמותה הישראלית להסמכת בודקים, Matrix Testing & Automation, ג'ון ברייס הדרכה, חברת כאל, Inflectra, Applause, Agile-i Trynagrub & Sparkly.io the SUT owner, Testing Days 2018.

התחרות מיועדת לבודקים מנוסים וגם לאלו החדשים בתחום, המשווגעים לדבר, אשר אוהבים אתגרים ונהנים לבצע בדיקות. במסגרת התחרות התבקשו המתמודדים לבדוק מוצר אמיתי בתנאים קיצוניים ובזמן מוגבל. המשתתפים נדרשו לזהות באגים, לדווח עליהם באופן ברור המאפשר תיקון, לערוך בדיקות לא פונקציונליות לפי הצורך ולסיום גם להגיש דו"ח סיכום מצב המוצר אשר יהווה ערך נוסף לחברת המוצר מבחינת שחרור הגרסה.

במוקדמות שנערכו ב-27 לאפריל 2018 השתתפו 24 צוותים ודווחו 419 תקלות.

בגמר שנערך ב-18 ליוני 2018 השתתפו 7 צוותים ונמצאו 132 תקלות.

צוות השופטים במוקדמות ובגמר כלל מנהלים ויועצי בדיקות בכירים המייצגים את חברות ה-IT וההייטק המובילות בישראל. אנה לוקובסקי (יו"ר ISTC וצוות השופטים), ירון צוברי, דני אלמוג, אלון לינצקי, דדי תנעמי וערן דמלין.

הזוכים של ISTC 2018

מקום ראשון

רועי אייל ואיגור גולדשמידט, מהנדסי בדיקות מחברת Gett

"הסיפור התחיל לפני שנה שאיגור שמע פעם ראשונה על התחרות בדיקות בעולם. ואז כשראינו את הפרסום בפייסבוק כל אחד מאיתנו התחיל לחפש שותף לתחרות. מיד חשבנו אחד על השני כידועים בתור "חולי בדיקות".

אז התחלנו להתכונן, לחקור חומרים שהמארגנים שלחו לנו, הסתכלנו בוידאו של תחרויות קודמות. אחרי שבוער הבנו מה נדרש לזכייה, התחלנו להתארגן עם כלים, מסמכים, צ'ק ליסטים לסוגי בדיקות כלליים וקבענו תהליך עבודה מתוזמן בתוך הצוות.

בגדול, הרעיון הכללי לניצחון היה להשקיע בלימוד, ניתוח מערכת וכתובת High Level Test Plan

ואז לבצע בדיקות.

בשלב ראשון של תחרות במהלך המפגש הבנו שהאסטרטגיה שלנו לא תעבוד ובמקום החלטנו ישר להתרכז בבדיקות אחרי הבנת המטרה ומהות המוצר.

בשלב השני כבר הגענו מוכנים והמשכנו עם האסטרטגיה הקיימת ורק שיפרנו בקטנה ביצועים אחרי הפיידבק של השופטים. אתגרים מרכזיים של התחרות הם לחץ של זמן ושמירת קור רוח. היה מאוד קשה לא להיתקע בדברים שלא עובדים ולהמשיך הלאה. כל פעם לבחור את העדפות הנכונות של ריכוז בדיקות בזמן הרצה ועמידה בלוח הזמנים שהכנו מראש. בסה"כ מאוד נהנינו מהתהליך, האתגר ועבודת הצוות. הכרנו אנשים מגניבים שגם חיים את עולם הבדיקות ומוצאים זמן לעשות מזה כיף.

שוב תודה למארגנים ולאנה שמייצרת הזדמנות להשתתף בחווית שכזאת."

מקום שני

שמואל גרשון בודק תוכנה באינטל ירושלים, **ומיכאל שטאל** ארכיטקט בדיקות תוכנה באינטל ירושלים

"אנו מאמינים באמונה שלמה שלמידה מתמדת וסקרנות הם מגדלור לבדוק תוכנות, וגם מרכיבים מצוינים לכיף בעבודה."

מקום שלישי

יהונתן דיוויס מהנדס בדיקות מ-Inherent Simplicity

"אני מאוד אוהב את תחום הבדיקות ומתנסה בכל אתגר אפשרי שאני מוצא בתחום. כששמעתי על התחרות, לא יכולתי להתעלם מהאתגר והרגשתי שאני חייב לפחות להתנסות ולטעום, מה זה בעצם תחרות QA וככה בעצם הגעתי לתחרות של ISTC.

אפשר להגיד שזה ה"אני מאמין" הכי גדול שלי: אני מאמין בכוחות הפנימיים שלנו כבני אדם.

כל אדם, לא חשוב מי הוא ומאיזה רקע הוא בא, יכול להגיע לאן שירצה, אם ירצה את זה מספיק ויהיה מוכן לעשות את כל המאמצים הדרושים. אני מאמין שאנחנו קובעים את הגבול של ההצלחה שלנו."





ברכות גם לשאר הצוותים שעלו והתחרו בגמר גביע זכר בתארים מיוחדים

Best Bug Award

דן קופייקין מהנדס בדיקות ואורי ברכה מפתח
אוטומציה מ-Payoneer

"אנחנו עובדים ביחד באותו הצוות כבר יותר משנתיים.
זכינו במקום השלישי בתחרות הבדיקות 2017 ורצינו להוכיח לעצמנו שהתקדמנו והשתפרנו במהלך
השנה האחרונה. מעבר לאתגר המקצועי, אנחנו נהנים ברמה האישית לקחת חלק בתחרות זו."



Best Security Testing Award

מיטל בן-צבי בודקת תוכנה Client-Server וקורי אבדין מנהלת צוות בודקי תוכנה Client-Server
מ-Blackberry

"אנחנו חברות טובות ועובדות יחד בצוות Client-Server בחברת BlackBerry.
הדינמיקה ביננו מעולה, נהננו מהאתגר בתחרות בדיקות בשנה שעברה והחלטנו להשתתף יחד שוב!
אנחנו דוגלות בחידושים, נותנות לסקרנות להוביל אותנו בכל בדיקה שאנו מבצעות, ומאמינות בלא
לקחת שום דבר כמובן מאליי.
כמו כן, חשוב לנו תמיד להישאר מעודכנות בטכנולוגיות חדשות, להיות מעורבות בהרצאות ולהשתתף
בפעילויות כמו התחרות הזו.
האני מאמין שלנו הוא להסתכל על המוצר מכמה פרספקטיבות כדי להגיע לתוצר הטוב ביותר בשביל
הלקוח, ובצורה היעילה ביותר."

Best Test Plan Award

מקס סקלט מנהל QA בחברת NiNiSpeech ואולגה כהן מובילה את תחום הבדיקות במחלקת
QA בספירל-סולושנס

"החלטנו להשתתף ביחד בתחרות ISTC 2018 מפני שעבדנו ביחד כמה שנים עד לפני כחודש, ויש
בינינו הסכמה מלאה על איך איש בדיקות טוב צריך לעשות את עבודתו.
אנחנו מאמינים שביחד אנחנו צוות מנצח, עצמאי וחזק. אנחנו משלימים אחת את השני בכך שכל אחד
מביא את התכונות החזקות שלו.
אנחנו מאמינים שאיש בדיקות טוב צריך להיות סקרן, חפזן (לשאול הרבה שאלות), מאוד יסודי, ייצרתי,
לדעת לעבוד בצוות וכמובן להיות ביחסי אנוש עם כל הקולגות ואנשים מסביב."

Best Test Scenario Award

רותם גורנר ראש צוות בדיקות מובייל ב-QualiTst וניר טל מהנדס QA בכיר בחברת vHive

"הכרנו והתחברנו מהאוניברסיטה, אוהבים את עולם התוכנה ומאמינים בחשיבותם של בדיקות התוכנה כהליך במחזור חיי
התוכנה משלביו המוקדמים ביותר.
אנו נמצאים בתהליך מתמיד של למידה ושיפור ושואפים לספק את התוצר הטוב ביותר ללקוחותינו!"

משתתפים	סטטיסטיקות המוקדמות	סטטיסטיקות הגמר
	24 צוותים, 15 זוגות, 9 צוותי יחיד	7 צוותים, 6 זוגות, 1 צוות של יחיד
המוצר הנבדק	Trynagrub an exclusive mobile dining platform for hungry foodies	Sparkly.io a new way for publishers, agencies and brands to tell their stories on mobile devices-
פרק זמן הבדיקה	117 שעות אדם	39 שעות אדם
כמות תקלות שדווחו	419	132
כמות ממוצעת של תקלות שנמצאו לצוות	18	18
מקסימום של תקלות שנמצאו ע"י צוות אחד	45	24
תקלות אמיתיות שנמצאו ע"י צוות אחד	22	18
מינימום של תקלות שנמצאו לצוות	2	11
אחוז Most Valuable Bugs מתוך סך התקלות	4.5%	7%



אייל זילברמן

מומחה בבדיקות בכיר ומנכ"ל קווליסט, חברת הבדיקות המובילה בישראל והשנייה בגודלה בעולם. פעיל בתחום בדיקות התוכנה משנת 1995, עבד כבודק תוכנה, מנהל ומוביל בדיקות במספר ארגונים כמו בנק ירושלים, T-Mobile, Teleknowledge ועוד. פרסם עשרות מאמרים מקצועיים ונאם פעיל בכנסים ישראליים ובינלאומיים. אייל הוא בין המקימים של מגזין בדיקות התוכנה הישראלי הראשון "חושבים בבדיקות" וחבר בצוות ה-AB של ITCB – ארגון ההסמכה לבדיקות הישראלי.

שילוביות רציפה - Continuous Integration

* למי שאין רקע ב-DevOps מומלץ לקרוא את הטור שהתפרסם בנושא בגיליון 1 של עולם הבדיקות <http://bit.ly/2J56M25>

האבולוציה של מתודות וגישות לפיתוח תוכנה לאורך השנים התמקדו בתכיפות שחרור הגרסאות, מחודשים (במפל Waterfall) לשבועות (אגילי לימים ושעות (DevOps). שחרור גרסה הוא אתגר אמיתי שמחייב כי התוכנה המופצת לשוק תהיה אמינה ועובדת בכל רגע נתון. שילוביות רציפה (Continuous Integration או CI) פותרת את האתגר באמצעות איחוד כל השינויים המבוצעים בתוכנה ואינטגרציה שלהם באופן שוטף, ולפחות פעם אחת ביום. בשילוביות רציפה המפתחים מכניסים קוד למוצר כל הזמן. ומה עושים הבודקים? בודקים כל הזמן. היתרון העיקרי מנקודת המבט של איכות הוא כמובן היכולת לבדוק מוקדם יותר ולמצוא תקלות מוקדם יותר. אחד העקרונות הבסיסיים בשילוביות רציפה הוא אוטומציה. כל מה שאפשר לעשות בצורה אוטומטית, נעשה בצורה אוטומטית. הרבה פעולות שבעבר בוצעו רק באופן ידני (כגון בדיקות יחידה וניתוח כיסוי קוד) נעשים בצורה אוטומטית. עקרון האוטומציה לא פוסח על הבדיקות. צוותי בדיקות העובדים בשילוביות רציפה כוללים כיסוי אוטומציה רחב יותר ויישום אוטומציה בתהליכים שבעבר בוצעו בצורה ידנית (לדוגמה: הרצת האוטומציה או דיווח תקלות למפתחים).

כלים

בעבר כלי ניהול תצורה (Configuration Management) היו מאוד פופולריים ועזרו לצוותי פיתוח לנהל גרסאות ולנהל את קוד המוצר במקום מרכזי. כלים אלו עברו אבולוציה, והיום כלי CI מנהלים את ניהול הסביבות והמשימות, אבל בזמן אמת ובעיקר כל הזמן. הכלי הפופולרי ביותר הוא Jenkins (שהוא כלי קוד פתוח) אולם קיימים כלים נוספים אשר הופכים ליותר ויותר פופולריים כגון Travis CI או Circle CI אשר מאפשרים אינטגרציה במודל אירוח בענן (Hosted).

איך זה עובד?

לאחר שהמפתחים כותבים קוד ומדבגים אותו הם עושים לקוד Check In לתוך קונטיינר של קוד מקור. תהליך CI כולל את הפעולות האוטומטיות הבאות (או חלקן):

- ניתוח קוד סטטי – ביצוע ניתוח קוד סטטי באמצעות כלי אוטומטי ודיווח תוצאות
- הידור (קומפילציה) – הידור ואיחוד הקוד
- בדיקות יחידה (Unit Test) – הרצת בדיקות יחידה, בדיקת כיסוי קוד (Code Coverage) ודיווח תוצאות
- התקנה של הגרסה על סביבת בדיקות
- הרצת בדיקות אינטגרציה ודיווח תוצאות
- עדכון בזמן אמת – הפצת סטטוס כל הפעולות לעיל לדשבורד ציבורי או שליחת דוא"ל לצוות הפיתוח והבדיקות

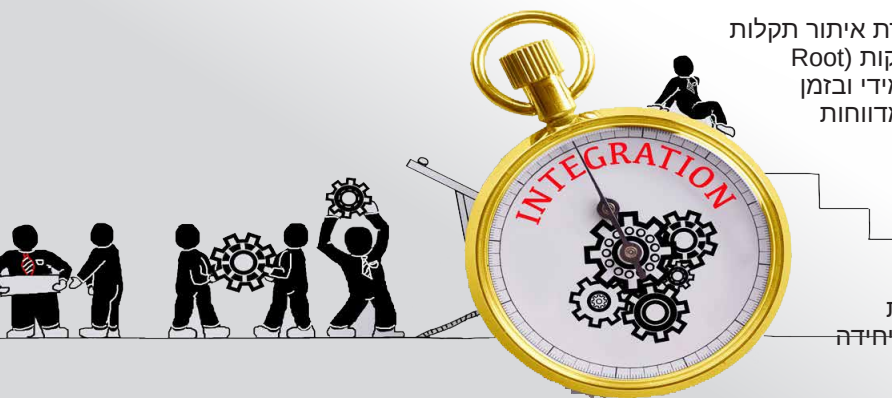
איך זה משפיע על צוות הבדיקות?

עבודת הבדיקות בשילוביות רציפה שונה מהותית מבדיקות בעולם המסורתי. להלן עיקרי השינויים הנדרשים מצוות הבדיקות במעבר לשילוביות רציפה:

1. בעבר צוות הבדיקות היה מתחיל בהרצת בדיקות מייד עם סיום הפיתוח. ובשאר הזמן עסק הצוות בפעילויות תחזוקה ותכנון שוטפות. בשילוביות רציפה צוות הפיתוח מקבל משוב מצוות הבדיקות באופן רציף לאורך כל מחזור הפיתוח.
2. אוטומציה הייתה בעבר משהו נחמד שעוזר ומייעל את תהליך הבדיקות. בשילוביות רציפה אוטומציה היא חובה.
3. אם בעבר אוטומציה פותחה ע"י מומחי אוטומציה ייעודיים, בשילוביות רציפה הבודק, ובמקרים רבים גם צוות הפיתוח, מחויבים לידע נרחב בעולם האוטומציה ויכולת לפתח ולהריץ סקריפטים אוטומטיים
4. אוטומציה לא מתמצה בהרצת בדיקות. כל התהליכים הסובבים את התוכנה, מהידור הקוד ועד פרישתו בשטח נעשים בצורה אוטומטית ככל שניתן.
5. שימוש תכוף יותר בכלי בדיקות קוד סטטי וכיסוי קוד כדי לנתח בצורה אוטומטית את הקוד בו נעשה שינוי ולהריץ הבדיקות באופן שוטף.

סיכום

שילוביות רציפה מייצרת יתרונות רבים לבדיקות. היא מאפשרת איתור תקלות בשלבים מוקדמים, מאפשרת ניתוח טוב יותר של מקור הבדיקות (Root Cause Analysis), נותנת לצוותי הבדיקות והפיתוח משוב תמידי ובזמן אמת, מצמצמת את הזמן מהפיתוח לבדיקות כך שהתקלות מדווחות למפתחים יום אחד אחרי סיום הפיתוח ולא שבועות וחודשים אחרי, מייצרת את רוב העבודה הסיזיפית של בדיקות הרגרסיה ומאפשרת כיסוי גבוה יותר של אוטומציה ועוד. לבודקי התוכנה המעבר לשילוביות רציפה הוא אתגר אמיתי ומחייב אותם להיכנס לתחומים כמו ניתוח קוד או יצירת בדיקות אוטומטיות שבעבר היו שייכים למומחים ספציפיים בתחום. דגש חשוב לצוותי בדיקות המיישמים שילוביות רציפה הוא לא להסתמך רק על בדיקות יחידה ולהמשיך לבצע בדיקות אינטגרציה ובדיקות מערכת.





לנה הרמן שמוליאן

מהנדסת בדיקות תוכנה בחברת צ'קמרקס. במקור למדה ביוכימיה ומדעי המזון אך אחרי עבודה בתחום הבינה שאינה רוצה להישאר בתחום ועשתה הסבת מקצוע לבדיקות תוכנה בחברה אקספריס. בתום ההסבה עבדה 4.5 שנים ב-HP שהפכה ל-HPE ולאחר מכן למיקרופוסקס. לאחר חופשת הלידה השנייה החליטה לחפש את האתגר הבא והגיעה לצ'קמרקס. לנה אוהבת מאוד להוביל דברים, אוהבת אתגרים ומאוד אוהבת את תחום הבדיקות.

עליונה, והבודק אומר שאצלו דברים אחרים הם בעדיפות עליונה ושהוא לא יכול לבדוק את הפיצ'ר הזה עכשיו, והם יתפנו לזה רק עוד שלושה שבועות. זה תמיד להתרוצץ בין ר"צ פיתוח ור"צ QA, ובסופו של דבר מה שמעניין אותי מהצד שלי, זה שהמשימה תסתיים.

המשימה מסתיימת כשנעשה QA ידני, כשנעשתה אוטומציה, כשיש Unit Test, כשהכל משולב ועובד. בשביל שזה יקרה צריך לעבוד בצמוד, מפתח ואיש QA ביחד.

שמתי לב שזה עוזר מאוד שהבודק עובד ממש צמוד למפתח, כלומר מפתח סיים משהו, איש QA יושב לידו וישר אומר רגע בוא נבדוק למה עשית את זה וכרגע פגעת במשהו אחר, זה מאוד עוזר לשים פוקוס על הדברים שאתה עושה ולא לרוץ קדימה.

עבודה ביחד, היא עבודה שהיא הכי יעילה.

בצוות אגילי שהוא יותר מגוון, אולי יותר קשה לר"צ שבא מאזור של פיתוח לנהל גם אנשי QA כי אין לו את הידע המקצועי להדריך.

אבל בסופו של דבר מאוד ברור שזה מאוד עוזר למפתח וגם לאיש ה-QA כי הוא לא מנותק מהפיתוח, הוא יודע מה קורה בפנים, הוא מבין מה שהמפתח עושה ואם באמת מה שהוא עושה זה נכון ויכול לעצור את הדברים בזמן".

החסרונות

- מי שמנהל את הצוות זה איש פיתוח, לרוב ללא הסתכלות והבנה של איש בדיקות וזה יכול לגרום לסדרי עדיפויות שלא תומכים QA. כלומר העדיפויות תהיה לסיים פיתוח של פיצ'ר ללא התחשבות בזמן שלוקח לבדיקות, ואין במקרה כזה ראש צוות בדיקות שיתערב לטובת הבודק.
- איש הבדיקות נאלץ להתמודד עם מצבים שבהם הוא צריך להתעקש על איכות ולא להקל ראש בעניין הבאגים וגם להסתדר טוב עם כל המפתחים כי בשיטת העבודה הזאת הם חברי הצוות שלו.
- יש פחות תפקידי ניהול בכירים אפשריים בתחום הבדיקות
- הנקודות השונות שדיברנו עליהן ביתרונות, יכולות להוות גם חסרון בחלק מהמקרים כי הן גורמות ליצירת תשתית פחות טובה לבדיקות ולמדידות.

דורו בניאן - מהנדס תוכנה בכיר וסקראם מאסטר בחברת צ'קמרקס, מסכם את היתרונות והחסרונות בצורה טובה

"QA בקטע המקצועי, ככל שהוא מעורב בפיצ'ר בשלב יותר מוקדם זה יותר יעיל ויותר נכון.

מה זה יותר מוקדם? אם פעם בודק התחיל לבדוק פיצ'ר רק לאחר שהוא מסתיים אז היום הנטייה שזה יהיה הפוך, עוד לפני שהפיצ'ר מפותח, QA יהיה חלק מהפיצ'ר, יהיה מעורב בשיבות design ויחשבו על הבדיקות עוד לפני.

הפיצ'ר נבדק כחלק מהספרינטים, זהו התהליך הנכון.

אגילי, אגילי, אגילי, כל מי שלא חי בשנים האחרונות בעולם מבודד ושייך לעולם ההייטק, מכיר את המושג ויודע על מה מדובר. אבל מה זה אומר מבחינתנו הבודקים?

ברוב הארגונים המיישמים מתודולוגיות עבודה מעולם האגילי ישנם צוותי אגילי שבהם הבודקים והפיתוח הם באותו צוות ומנהלים ע"י ראש צוות פיתוח.

בעצם הבודק עדיין צריך לבדוק את המוצר בצורה הכי טובה שיש, לעבוד לפי מתודולוגיות עבודה של QA, לתכנן את כל מחזורי הבדיקות וסוגי הבדיקות רק ללא מוקד ידע שיוכל להדריך אותו בכל מה שקשור לבדיקות (רוב ראשי הצוותים בפיתוח לא יודעים לתת מענה לכך בינתיים, יכול להיות שעם הזמן ככל שיעבדו יותר שנים בשיטה זו יצברו את הידע הדרוש). כל זה משנה קצת את מה שנדרש מאיש הבדיקות להיות. לרוב לא קיימת לכך הכשרה שונה אבל הבודק בצוות כזה נדרש להיות יותר אסרטיבי, יותר ממוקד מטרה, עצמאי, מוביל, סוג של פוליטיקאי, וכמו כן מנוסה בתחום הבדיקות ולכן לבודק מתחיל יהיה קצת קשה להשתלב בצוות כזה (אלא אם כן הצוות כולל גם בודקים וותיקים שיוכלו להנחות אותו).

חשוב לציין שאומנם לא נדרשת הכשרה שונה אך בהחלט הכשרה ניהולית הופכת למשמעותית, מכיוון שבודק כזה לעתים נדרש לקבל החלטות שהיה מקבל ראש צוות QA ולראיין מועמדים לתפקיד הבדיקות בצוות.

כמובן שכמו לכל דבר, למתודולוגיית עבודה זו ישנם יתרונות וכמובן חסרונות. בואו נסתכל על חלק מהאספקטים השונים של מתודולוגיית עבודה זו.

היתרונות

- **מענה מהיר יותר** - העבודה הצמודה מול הפיתוח מאפשרת לפתור בעיות שאנשי הבדיקות נתקלים בהם בזמן חקירה של באגים וכדומה.
- **תיקונים בשלב המוקדם** - העבודה הצמודה והתקשורת המתמדת בין הבודקים למפתחים מאפשרת לגלות דברים כמו אי התאמה בין הדרישות ליישום של המפתחים בשלב מאוד מוקדם, לפעמים בעקבות שיחת חולין בין חברי הצוות.
- **קיצור "הבירוקרטיה"** - לא נדרשת עכשיו התנהלות אינסופית של תיאומים בין ראשי הצוותים של הבדיקות והפיתוח.
- **מסירה מהירה ויעילה יותר של גרסאות**
- **אפיקי התפתחות חדשים** - בגלל המבנה הארגוני החדש, לאנשי בדיקות יש אפיקי התפתחות חדשים שלא היו פתוחים בפניהם במתודולוגיות עבודה אחרות, למשל איש בדיקות יכול להיות סקראם מאסטר של הצוות, QA Lead (תפקידו להוביל את הבדיקות מבחינת תהליכים ומתודולוגיות) ועוד.
- **צבירת ניסיון בפיתוח** - מכיוון שהבודקים נמצאים בתוך צוות הפיתוח פעמים רבות הם מקבלים משימות אוטומציה ומשימות שקשורות לקוד וכך יכולים ללמוד ולצבור ניסיון בפיתוח אוטומציה ובפיתוח בכלל.

אלכס רייכמן - מנהל קבוצה בחברת צ'קמרקס מחזק נקודות אלו,

"מהניסיון שלי (ויש לי ניסיון של הרבה שנים בתחום), כשעבדנו בוואטרפול תמיד הייתה בעיה שהמפתח עושה את מה שהוא עושה ומבחינתו המשימה הסתיימה כשהוא סיים את הפיתוח. ה-QA עובדים לגמרי במנותק מהפיתוח, יש להם סדר יום אחר, יש להם סדרי עדיפויות אחרים, לא פעם קורה שרק לאחר חודש, איש QA מגיע לבדיקות ואז לא הכל עובד, פונים למפתח אבל הוא כבר במשימה אחרת וכבר לגמרי שכח על מה עבד. אופן עבודה כזה יכול להיות בעייתי.

איש QA צריך לעבוד בצמוד עם המפתח, אחרת נוצר פער יותר מדי גדול בזמנים.

לפעמים קיים ניהוג אינטרסים. המפתח אומר שזה אצלו בעדיפות



דבר נוסף שיכול להדאיג דווקא את המנהלים שבינינו הוא העתיד של מנהלי הבדיקות בחברות שעובדות במודל העבודה המדובר.

האם תפקידי הניהול באמת נעלמים? ומה מנהלי הבדיקות יכולים לעשות?

כפי שצינתי, מודל העבודה הזה מוליד הזדמנויות חדשות לבודקים: ברוב החברות יש תפקיד חדש שהוא QA Lead.

כמו כן בודקים יכולים לפנות לכיווני התפתחות קצת שונים כמו סקראם מאסטר, מנהל פרויקט, מנהל מוצר וכו'.

לכן אני בתור בודקת תוכנה לא דואגת, כמו שאומרת [נוגה יהודאי](#) ר"צ QA בחברת Akamai

"אני חושבת שלא מחויב המציאות שתפקידי ניהול הבדיקות יפגעו, לפחות לא ברמה של ר"צ. אבל אם מסתכלים על ההיררכיה של ה-QA זה כמעט נעלם מהעולם (מצב שבו יש דירקטור QA מתחתיו מנהל מחלקה ומתחתיו ר"צ). אם נתמקד בתפקיד ר"צ QA אז התפקיד הזה משנה פנים. העיסוק הוא יותר בהגדרה של מה ואיך נכון לבדוק, מערכות אוטומציה מתקדמות וכמובן הפן האנושי של ניהול כ"א, הובלת תהליכים וקשר הדוק עם DevOps ופחות בניהול המשימות היומיומיות (שעל זה אחראי ה-Scrum Master).

אני חייבת לציין שבמקומות רבים החליטו לוותר על הפונקציה של ר"צ בדיקות ולדעתי זו החלטה שגויה (כמו גם בעיני ארגונים רבים שהחליטו החלטה מודעת שלא לוותר על פונקציה זו). התפיסה של אוטומציה=QA הינה טעות. אוטומציה כל מפתח יכול לכתוב אבל בדיקות טובות שבאמת יתקילו את המערכת - בשביל זה צריך אנשי בדיקות טובים והובלה טובה שתדע לדרוש את זה מהם."

בנוסף ישנן חברות שבהן מאמצים אג'יל בצורה קצת שונה ומשאירים הפרדה בין הצוותים, כלומר יש צוות פיתוח עם ראש צוות פיתוח וצוות בדיקות עם ראש צוות בדיקות והצוותים עובדים במקביל. אז כמו שנוגה אמרה תפקיד ר"צ QA לא נעלם לגמרי.

לסיכום, אומנם חלק מאיתנו אולי חושש אך תפקיד הבודק לא נעלם, הוא רק משנה פנים.

ההיררכיה קצת משתנה, תהליכי העבודה אולי אחרים, יש לבדוק התמודדויות חדשות אבל השורה התחתונה זהה, לא משנה באיזה מתודולוגיה עובדים ולא משנה אם אתה בודק או מפתח

לכולנו יש אינטרס משותף שהוא לספק מוצרים עובדים באיכות גבוהה, בעלי יתרון תחרותי בשוק ובעלי ערך מוסף למשתמש.

בקטע האישי זה טיפה יותר טריקי, מצד אחד יש מנהל צוות שנותר לבדוק את המשימות, אך האחראי עליו מקצועית יהיה QA Lead, ז"א, מצד אחד הוא מנהל מקצועית ע"י מישור אחד וניהולית ע"י מישור אחר.

עבדתי גם במתודולוגיות שונות לגמרי, למשל שמנהל פיתוח מנהל רק את הפיתוח ומנהל QA הוא מישור אחר לגמרי ואינו כפוף למנהל הפיתוח.

מתודולוגיה כזאת מונעת מצב שבו מנהל פיתוח יגיד - טוב ה-QA שלי אז אני מאשר שזה בסדר. אז לא, הוא אחראי על הפיתוח ומישור אחר לגמרי, גוף אחר, עם מנהל אחר בודק אותו.

מה בעצם מונע ממנהל הפיתוח עכשיו להגיד אני קובע שהכל בסדר? הרי בשביל שפיצ'ר יהיה מוכן הוא צריך לעבור את כל המסלול, אין טעם להגיד שהפיצ'ר מוכן אם לא עבר בדיקה של QA.

מבחינה מקצועית השיטה שהבודק בתוך הצוות יותר יעילה אך בפן הניהולי זה יותר קשה ונדרש מאיש ה-QA להיות הרבה יותר חזק."

עוד נקודה למחשבה כשמדובר על מתודולוגיית העבודה הזאת היא איך רואים המפתחים את שיטת העבודה הזאת. כדי לענות על השאלה הזאת נראה לי הכי טוב שאביא כאן כמה ציטוטים של מפתחים שיצא לי לעבוד איתם:

[סיון אגרנוב](#) - web fullstack developer בחברת Glispa

"אני חושב שהמעגל היותר קצר של פיתוח<בדיקות<מסירה הוא מעולה, כלומר לסיים תכולה, לתת אותה מייד ל-QA ולא לחכות לאיזה תאריך ספציפי שקבעת עם הצוות QA שבו הם מתחילים לבדוק."

[יונתן אונגר](#) - מפתח תוכנה בחברת צ'קמרקס

"שיטת העבודה הזו מרגישה לי טבעית. ה-QA הוא חלק מתהליך הפיתוח, לי טבעי שכל פיצ'ר חדש שאני מכניס יוכל להיבדק מהר ובצורה ששקופה לי. גם ל-QA זה שקוף, אני יכול להסביר ל-QA מאוד מהר. גם לי שקוף אם משהו נופל, זה לא שזה מישור בצוות שאני לא מכיר או צוות שלא קשור אליי בודק. זה לא ה-QA נגדי, זה שנינו ביחד."

אז נראה שדווקא המפתחים מקבלים את שיטת העבודה הזאת ברוח טובה וחושבים שהיא עדיפה.





דם, יזע ודו"ח ביצועים

הרצה של בדיקה פונקציונלית מסתיימת בתוצאה סופית ברורה: עבר או נכשל. דיווח על בדיקות ביצועים (performance) לעומת זאת, זה משהו אחר לגמרי.



מיכאל שטאל

הוא ארכיטקט בדיקות תוכנה באינטל, ישראל. במשך 17 השנים האחרונות מיכאל בדק מוצרים בתחום Wi-Fi, טלוויזיות חכמות, כרטיסים גרפיים, אפליקציות ראייה ממוחשבת מבוססות על מצלמות תלת-מימד, ולאחרונה הוא עובד בקבוצה שבודקת את התוכנה שמאחורי טכנולוגיית ניהול אקטיבי ומנוע האבטחה (CSME) של אינטל. במסגרת תפקידו, מיכאל מגדיר שיטות בדיקה ומתודולוגיות עבודה, עוסק הרבה בהדרכה ולפעמים אפילו מרשים לו לבדוק משהו (שזה הכי כיף). מיכאל מציג תכופות בכנסים בארץ ובחו"ל והציג בין היתר ב-SIGIST Israel, STAR conferences, QA&Test, ובכנסים אחרים. מיכאל מלמד בדיקות תוכנה בפקולטה למדעי המחשב באוניברסיטה העברית. ניתן לראות חלק מהמצגות והמאמרים שלו באתר

www.testprincipia.com

קודם כל הגדרה:

תחת השם "בדיקות ביצועים" אני מכניס, לצורך המאמר הזה, כל בדיקה שהתוצאה שלה היא מדידה של משהו; ערך מספרי שיכול לנוע בתוך תחום מסוים ועדיין להיחשב כתקין. זה יכול להיות מדידת צריכת זרם, כמות המשתמשים שאתר רשת מסוגל לטפל בה בו-זמנית, מהירות שלילפת נתונים מהדיסק, וכו' וכו'. למעשה: כל מדידה של דרישה לא-פונקציונלית.

הקושי הראשון של בדיקות ביצועים הוא לקבוע מה טווח התוצאות שיחשב "כעובר". זה מתחיל כבר בשלב הגדרת הדרישות. הרבה פעמים הדרישה היא משהו בסגנון הבא: "זמן אחזור נתון מבסיס הנתונים לא יעלה על 10 מילישניות" או "המערכת תעבד קבצי וידיאו בקצב של לפחות 100 תמונות בשנייה". הגדרות כאלה הן חסרות (incomplete) כי לא ברור מה התוצאה הרצויה. ידוע רק מה התוצאה הגרועה ביותר שנמדוד ועדיין נאמר שהמוצר עבר את הבדיקה. למעשה מתחבאות כאן שתי בעיות:

(1) נניח שאני מריץ בדיקה ומוצא שקצב עיבוד התמונות הוא 101 תמונות לשנייה (הדרישה הייתה "לפחות 100 תמונות בשנייה"). נראה טוב, נכון? אבל האם זה אומר שאני קרוב לקצה (כלומר, עברתי את הבדיקה בעור שיני) או שהכל תקין? אם הדרישה הייתה מוגדרת היטב, היא הייתה כוללת את המטרה (נאמר: 120 תמונות לשנייה) וגם את מינימום הביצועים הנדרשים: 100 תמונות לשנייה. עכשיו, כשאני מודד 101 תמונות לשנייה, ברור שהמוצר עומד אך בקושי בדרישה.

(2) כאשר נכשלים בבדיקה רק קצת (נגיד, 99 תמונות לשנייה), יש לחץ על מנהל המוצר "להתגמש". "אוקי, זה באמת מתחת לגבול, אבל הרי אנחנו כמעט עוברים, אז אפשר להחליט שזה בסדר". אילו הדרישה המלאה הייתה עומדת מול עיני המחליטים, הם היו מודעים לעובדה שהמוצר רחוק מהמטרה ושיש כאן בעיה אמיתית.

רק לשם שלמות, אזכיר שהגדרת דרישה לא-פונקציונלית צריכה לכלול בפרוטרוט גם את תנאי הבדיקה ומהלכה, שכן אלה משפיעים על התוצאה הנמדדת (למשל, בבדיקת עומס על המעבד: האם אנחנו מודדים Max CPU Utilization או ממוצע לאורך זמן? כמה זמן? כל כמה זמן דוגמים את עומס המעבד? מה עוד רץ על המחשב במקביל?).

כל זה הוא רק הקדמה לנושא האמיתי של הטור.

איך מדווחים על בדיקות ביצועים?

עקרונית, מה הבעיה? מציגים את התוצאה ומציינים אם היא עברה או נכשלה. אבל מעבר לתוצאה אנו גם רוצים להראות כמה אנחנו קרובים למטרה, ולצורך כך צריך דיווח שמצד אחד אינו מסורבל ומצד שני מעביר למתבונן תמונה מלאה יותר של המצב. אפשר למשל להשתמש בטבלה:

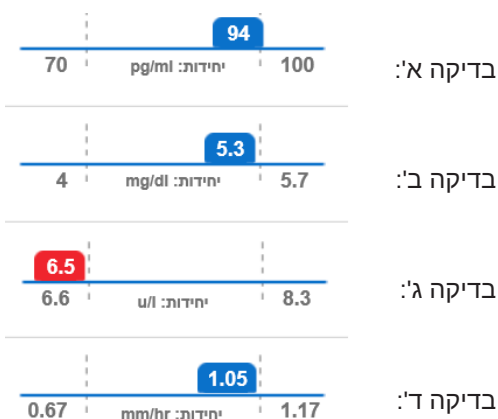
דרישה	מטרה (target)	מינימום (limit)	תוצאה
קצב עיבוד תמונות (fps)	120	100	101

זה אכן מספק את המידע, אבל כיוון שיש בכל מוצר יותר ממדידה אחת, הרי שנגיע לטבלה גדולה מלאה מספרים, וקשה יהיה להבחין בהסתכלות מהירה איפה מתחבאת בעיה. אפשר להשתמש בצבעים על מנת לשפר את הקריאות:

דרישה	מטרה (target)	מינימום (limit)	תוצאה
קצב עיבוד תמונות (fps)	120	100	101
CPU utilization	7%	10%	8.55%
צריכת הספק	1.5W	1.9W	1.34W

נראה לכם? למה קצב עיבוד תמונות קיבל אותו צבע כמו המדידה של ה-CPU? הרי אחד ממש קרוב לקצה והשני בכלל לא... מצד שני, האם יהיה מוצדק לצבוע את קצב עיבוד התמונות באדום? אם נעשה ככה, אז באיזה צבע נסמן כישלון בבדיקה? אדום חזק יותר? ומה ירוק אומר? עד מתי תוצאה היא ירוקה ומתי תהפוך לכתומה? בקיצור, צבעים קצת עוזרים, אבל גם מכניסים בעיות משלהם.

העניין הסתובב לי הרבה זמן בראש. גם כשרופא המשפחה שלח אותי לסבב בדיקות הדם השנתי, אותו אני מקפיד לעשות בערך פעם בשלוש שנים. התוצאות שחזרו מהמעבדה היו רשימה של כמה עשרות מספרים שאני מבין בהם פחות מתרגול במרק פירות. אבל... וכאן מגיע הקשר לענייננו: למרות שאינני רופא, יכולתי לומר מיד איפה יש תוצאה טובה, איפה גבולית ואיפה משהו שכדאי לשאול עליו את הרופא. ככה זה נראה:



מיד נדלקה נורה בראשי: הרי זה בדיוק האתגר בדיווח תוצאות בדיקות ביצועים! אולי אפשר לגנוב את השיטה וליישם גם אצלנו? לקחתי כמה נתונים ושיחקתי קצת עם PowerPoint, והרי התוצאה:



“איך לדווח תוצאות מוקדמות של בדיקות ביצועים, ובמקביל להעביר את המסר שזה תוצאות ראשוניות”

בדרך כלל בתחום של סטיית תקן אחת. לעיתים רחוקות יותר בתחום שבין 1 ל-2 סטיות תקן וכמעט אף פעם

לא בתחום שבין 2 ל-3 סטיות תקן. בכל מקרה, המדידות יתפלו באופן שווה פחות או יותר מעל ומתחת לממוצע. אם המדידות על גרסאות חדשות נופלות בתחום של סטיית תקן אחת, אבל באופן קבוע מעל (או מתחת) לממוצע, כנראה שאכן קרה שינוי. כמה פעמים המדידות צריכות להיות מוטות לכיוון מסוים על מנת שנוכל לומר שהשינוי משמעותי? זה תלוי בהתפלגות שלנו, ונכנס לתחום הנקרא "בקרת איכות סטטיסטית" (statistical process control). דוגמה לאלגוריתם המחליט מתי הטיה היא משמעותית הם הכללים של "ווסטרן דיגיטל":

- 8 וורסיות עוקבות של התוכנה שבהם הנטייה היא בתחום סטיית תקן אחת – אבל לאותו כיוון
- מדידות 5-5 וורסיות עוקבות, שמתוך 4 מדידות נותנות תוצאה בין סטיית תקן 1 ל-2 מעל (או מתחת) לממוצע – באותו כיוון
- מדידות 3-3 וורסיות עוקבות, שמתוך 2 מדידות נותנות תוצאה בין 2-3 סטיות תקן מעל (או מתחת) לממוצע – באותו כיוון
- מקרה יחיד שבו המדידה היא מעבר ל-3 סטיות תקן מהממוצע – לאחד הכיוונים

הצצה בוויקיפדיה מראה גרפים המתארים את הכללים האלה- (https://en.wikipedia.org/wiki/Western_Electric_rules).

כל זה טוב ויפה, אתם אומרים, אבל אנחנו לא יודעים מה היכולת של המערכת שלנו ואין לנו זמן להריץ מדידות חוזרות רבות על מנת לייצר את הסטטיסטיקה הזאת. לא תמיד אנחנו מריצים את הבדיקה על אותו מחשב ממש, כך שנכנס רעש נוסף שנובע מההבדלים הקטנים בין מחשב למחשב – גם אם המפרט שלהם זהה לחלוטין. במקרים שבהם כן יש מושג על ההתפלגות, יתכן מאוד שהיא אינה נורמלית ולכן לא ברור לפי איזה כללים יקבע מתי יש שינוי אמיתי.

מה עושים? במקרה כזה אפשר להחליט מתוך הבנת המערכת איזה שינוי נראה לכם משמעותי (נניח "שינוי של 2% הוא משמעותי"). עם החלטה כזו, אפשר להגדיר כללי אצבע פשוטים יותר ל"מגמה". הנה גישה שפיתחתי עם גל שטיינר, חבר לעבודה:

- (1) "מגמה" נקבעת על פי השוואה של התוצאה שנמדדה על הגרסה הנוכחית לעומת התוצאות של הגרסה הקודמת (גרסה N-1)
- (2) מגמה יכולה להיות: ללא שינוי, עלייה, ירידה
- (3) אם השינוי מ-N-1 הוא מעל ל-2%, יש מגמת עלייה
- (4) (או ירידה – תלוי בכיוון השינוי)

אם השינוי מ-N-1 הוא קטן מ-2% אזי נבדוק מה הייתה המגמה שנקבעה עבור התוצאות של וורסיה N-1:

מגמה נוכחית (וורסיה N)		המגמה שנקבעה בוורסיה N-1
עלייה של פחות מ-2% ביחס ל-N-1	ירידה של פחות מ-2% ביחס ל-N-1	
עלייה	ללא שינוי	עלייה
ירידה	ללא שינוי	ירידה
ללא שינוי	ללא שינוי	ללא שינוי

הגישה הבסיסית שהטבלה מממשת:

Feature	Score
Power consumption	97
Transactions/mSec	66
CPU utilization	24
Memory usage	44

שימו לב שאני עדיין משתמש בצבעים אבל הצביעה של קו התוצאות מסבירה את בחירת הצבע. בנוסף, היא מסבירה האם תוצאה גבוהה היא המועדפת או שמא התוצאה נמוכה. הצופה רואה בעצמו מה המשמעות, והצבע משמש בעיקר להפנות את תשומת הלב למה שאינו ירוק. כמובן שיצירה של דו"ח כזה לוקחת קצת זמן, אבל ודאי אפשר לאטמט את זה. אני בטוח שבמעבדה בקופת חולים לא יושב מישהו ומעדכן את תוצאות בדיקות הדם באופן ידני...

בנוסף לדיווח התוצאה, יש עוד שתי נקודות שצריך לפתור:

- איך התוצאה שקיבלנו בוורסיה הנוכחית משתווה לתוצאות קודמות? האם אנחנו במסלול שיפור או מתדרדרים? במילים אחרות מה המגמה (trend)?
- בדיקת ביצועים מחייבת לעיתים הכנת כמויות אדירות של נתונים (למשל בבדיקת ביצועים של בסיס נתונים; או דיוק של זיהוי עצמים מתוך תמונות) יתכן שבתחילת הפרויקט אין לנו עדיין מספיק נתונים, ומצד שני כבר התחלנו להריץ בדיקות ביצועים. איך לדווח תוצאות מוקדמות של בדיקות ביצועים, ובמקביל להעביר את המסר שזה תוצאות ראשוניות, מבוססות על נתונים חלקיים, ויתכן שישתנו משמעותית בעתיד?

חישוב המגמה

נניח שאנחנו מודדים את העומס על המעבד (CPU utilization). ירידה בעומס תצביע על שיפור ביעילות המוצר. אם התוצאה שמדדנו על הגרסה הנוכחית טיפה נמוכה מהערך שמדדנו על הגרסה הקודמת, האם נוכל לומר בוודאות שיש שיפור בגרסה החדשה? מסתבר שלא תמיד.

הרצת בדיקת העומס פעמים רבות על אותה וורסיה של התוכנה ועל אותה חומרה, תיתן פיזור מסוים של התוצאה. הסיבה לכך היא שבמקביל לקוד שלנו רצות במערכת עוד תוכנות רבות: שירותי מערכת ההפעלה, אנטי וירוס, כרטיס תקשורת שמגיב למסרים על הרשת, וכו' וכו' (בזמן שאני כותב שורות אלה רצות ברקע על המחשב שלי כ-150 תוכנות ושירותים). גם טמפרטורת הסביבה, רמת הבהירות של המסך ומידת הטעינה של הבטרייה משפיעות קצת על הביצועים. כל זה גורם לשונות במערכת הכוללת, ומכאן לשונות במדידות של ביצועי המעבד. בהרבה מקרים מכנים זאת "הרעש" שיש במדידה. כל זמן שהתוצאות של מדידה עדיין בתחום של הרעש, אי אפשר להגיד בוודאות שיש מגמה של שינוי.

אם נריץ את הבדיקה פעמים רבות, נוכל לחשב התפלגות של התוצאות ולקבל מושג על "היכולת של המערכת" (system capability). אם התוצאות מתפלגות לפי התפלגות מוכרת, אפשר לומר ברמה סבירה של בטחון מתי תוצאה אכן מראה על שינוי אמיתי. לדוגמה: אם ההתפלגות נורמלית (גרף פעמון) כל תוצאה שנופלת בתוך תחום של +/-3 סטיות תקן מהממוצע אינה בוודאות שינוי.

מתי כן נוכל לומר שיש שינוי? בכמה מקרים:

- אם המדידה חורגת מתחום ההתפלגות הרגיל (בהתפלגות נורמלית זה יהיה מעבר לתחום 3-3 סטיות תקן מהממוצע)
- אם המדידה עדיין בתחום ההתפלגות, אבל מדידות על וורסיות חדשות מראות נטייה עקבית לאותו כיוון. בהתפלגות נורמלית, אם אין שינוי ביעילות הקוד, תוצאות של מדידות עוקבות יפלו



אם כיוון השינוי (הקטן) בוורסיה N ממשיך באותו כיוון של שינוי משמעותי שקרה בוורסיה N-1, הרי שהשינוי ממשיך להיחשב משמעותי. אם יש שבירת כיוון, אז השינוי אינו נחשב משמעותי. זה מימוש פשטני ביותר ועקב כך אינו מושלם. למשל, הוא מאפשר לשינוי קטן אך מונטוני לחמוק ללא השגחה (עליה במדידה בכל וורסיה, אבל בפחות מ-2% כל פעם). אפשר לשכלל את הגישה על ידי הסתכלות גם על התוצאות של וורסיה N-2, או חישובים של EWMA (ממוצע משוקלל נע).

https://en.wikipedia.org/wiki/EWMA_chart

אפשר ורצוי להוסיף את המגמה לתצוגה של התוצאות (בדוגמה זאת גם הוספתי עמודה עם התוצאות של המדידות – כהדגמה לווריאציה אפשרית על הרעיון הבסיסי):

דיווח על תוצאות ראשוניות

כאשר אתם בתחילת הפרויקט, יתכן וטרם מימשתם את כל הבדיקות שעל פיהם תקבעו את ביצועי המערכת. אתם כבר מריצים קצת בדיקות ביצועים ומעוניינים לפרסם את המידע שיש לכם אבל אתם חוששים. ידוע לכם שהתוצאות הן ראשוניות, וכשכל הבדיקות יפותחו ויורצו יתכן שהתוצאה תהיה שונה באופן משמעותי מהתוצאות הראשוניות. איך אפשר, מצד אחד לדווח, ומצד שני להעביר מסר ברור שהתוצאות ראשוניות ויתכן שישתנו בהמשך באופן קיצוני? רק להגיד את זה לא תמיד עוזר. אנשים יצטוו את המספר שדיווחתם – לא את המלל וההסתייגויות שליוו אותו.

ניקח דוגמה: אתם מפתחים מערכת אוטומציה לאריזה של פלפלים. המערכת משתמשת במצלמה ובאפליקציית עיבוד תמונה על מנת לזהות פלפלים פגומים ולשלוח אותם מפס האריזה. על מנת לבדוק את המערכת

איך נדע האם אנחנו במסלול שיפור או מתדרדרים? מה המגמה?

אתם מתכננים להקליט את הווידאו הנקלט במצלמה כאשר על הפס עוברים פלפלים טובים מעורבים בפלפלים עם פגיונות שונות. סרטים אלה יורנו במעבדה לאלגוריתם שמזהה פלפלים פגומים ואתם תבדקו כמה מהפגומים זווה, וכמה מהפלפלים הטובים זווה בטעות כפגומים. הקלטות אלה לוקחות זמן רב, ועד עכשיו הקלטתם רק 60 מתוך 200 ההקלטות המתוכננות. הרצתם את ההקלטות דרך המערכת, וקיבלתם שרמת הזיהוי של הפלפלים הפגומים היא 75%. לדווח או לא? הרי יכול להיות ששאר ההקלטות יורו על תוצאות טובות יותר, ואז סתם יצרתם פאניקה. מצד שני יכול להיות שהתוצאות על שאר ההקלטות יהיו גרועות, ואז יצרנו הרגשת בטחון שאינה במקומה והפתעה לא נעימה בהמשך.

הפתרון הוא לדווח, אבל בצורה שתעביר מסר של אי וודאות. אני עושה כאן שימוש של רעיון שנקרא "cone of uncertainty" – https://en.wikipedia.org/wiki/Cone_of_Uncertainty

בתור התחלה, נסתכל על התפלגות התוצאות שקיבלנו עבור 60 ההקלטות שכבר הרצנו. לצורך הדוגמה נאמר שהסתבר לנו שב-10% מההקלטות, זוהו 58% או פחות מהפלפלים הפגומים. ואילו ב-10% מההקלטות זוהו 93% או יותר מהפגומים. כלומר שטווח הזיהוי בהקלטות שבין העשירון התחתון לעליון הוא זיהוי של 58% - 93%. נעשה עכשיו הנחה שהתפלגות זו מגדירה פחות או יותר את יכולת המערכת שלנו, ושבהקלטות הבאות אלה יהיו התוצאות הקיצוניות האפשריות.

- במקרה הטוב ביותר, כל 140 ההקלטות הבאות יראו זיהוי של 93%

- במקרה הרע ביותר, כל ההקלטות יראו זיהוי של 58%

מה אם כך התוצאה האפשרית הסופית, בהתבסס על הסטטיסטיקה שבידנו כרגע, ובידיעה שיהיו עוד 140 הקלטות?

Feature	Result	Score	Trend
Power consumption	97%		➡
Transactions/mSec	66 t/sec		➡
CPU utilization	24%		➡
Memory usage	44 MB		➡

במקרה הרע:	במקרה הטוב:
$\frac{(140 \cdot 58\% + 60 \cdot 75\%)}{200} = 63\%$	$\frac{(140 \cdot 93\% + 60 \cdot 75\%)}{200} = 87\%$

הדיווח אם כך יהיה:

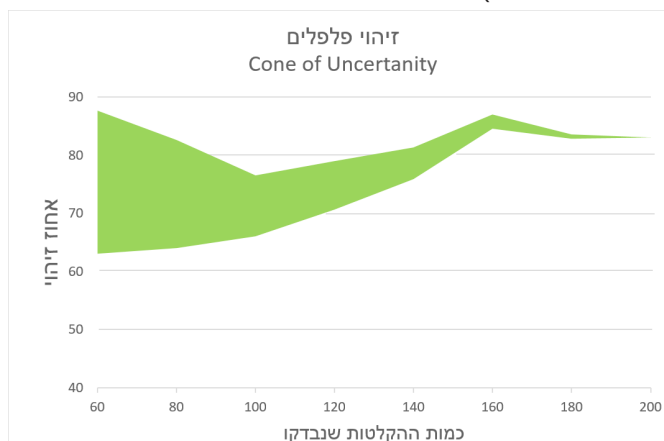
פרמטר	טווח תוצאה צפויה	כמות הבדיקות שבוצעו
זיהוי פלפלים פגומים	63% - 88%	30% [60 מתוך 200]

ככל שנתקדם בהקלטות ובהרצת הבדיקות, יהיו לנו יותר תוצאות אמיתיות ופחות אי וודאות לגבי התוצאה הצפויה בסוף. למשל, אם אחרי 120 הקלטות העשירון התחתון מראה זיהוי של 67%, העשירון העליון זיהוי של 88% והממוצע הוא 73%, הערכת הטווח תהיה:

במקרה הרע:	במקרה הטוב:
$\frac{(80 \cdot 67\% + 120 \cdot 73\%)}{200} = 70.6\%$	$\frac{(80 \cdot 88\% + 120 \cdot 73\%)}{200} = 79.0\%$

שימו לב שטווח אי הוודאות הצטמצם: 8.4% לעומת 24.5% אחרי 60 בדיקות.

ככל שנתקרב להרצה המלאה של כל הבדיקות המתוכננות, טווח אי הוודאות ירד. גרפית זה יראה משהו כזה (ועכשיו גם ברור למה זה נקרא "קונוס אי הוודאות" – הגרף נראה כמו קונוס שהולך וצר ככל שיש לנו יותר מידע):



גם אם זיהוי פלפלים הוא לא תחום העיסוק שלכם, אני מניח שיש מדידות שאתם עושים שהדיווח עליהם הוא ממוצע של בדיקות רבות. בכל המקרים האלה, הרעיון של קונוס אי וודאות יכול להיות רלוונטי ומאפשר דיווח חלקי, אך נכון, כבר בשלבים מוקדמים.

ועכשיו קצת Reality Check: חוץ מסימון המגמה, שאר הרעיונות שפרסמתי כאן הם עדיין בגדר רעיונות. כלומר: לא ראיתי מישהו שמימש אותם בפרויקט אמיתי. אני מפרסם אותו כאן בתקווה שמישהו מכאן יוכל להשתמש בזה.

ספרו לי אם זה עזר לכם!





הקשב תחילה – ורק אז קבע עמדתך



קובי הלפרין - עוסק בבדיקות מזה מעל ל-20 שנה בעיקר בתחום התקשורת טלקום/דאטהקום (ECI, Alvarion, Axerra, Ceragon), בעל מעורבות גבוהה בקהילות בעולם ובארץ ומוביל מספר פורומים אינטרנטיים, מחפש להרחיב את הדיון בתחום הבדיקות, שיפור כלים והרחבת הידע של הבודקים. פעיל בוועד המייעץ של ITCB® – בעיקר בשיפור האתר www.itcb.org.il והקשרים ברשתות החברתיות.



"הקשב תחילה – ורק אז קבע עמדתך" – כותרת זו החזירה אותי כ-15 שנה אחורה, לנקודה אשר נחרטה בראשי - בה אחת מראשי צוות הפיתוח אמרה לי בעודנו יוצאים מעוד אחת משיבות הובלת הגירסה (CCB) – "אנחנו ממחרים לתת תשובה שלילית – עוד בטרם שקלנו לעומק",

איני יודע אם זהו חלק מטבע האדם, ואולי חלק מהווי חיינו כמפתחים בהייטק בישראל בו מהירות התגובה נראית כחשובה יותר מהשיקול המסודר - בדומה להרגלינו מהצבא.

אולי בשל הטיות ("דעות קדומות") שאנו מחזיקים ומשפיעות על שיקול דעתנו מבלי משים, ואולי פשוט נבע מתופעת ה"סג"מ' - התחושה של ראשי צוות צעירים כאילו מעצם המינוי

אנחנו ממחרים לתת תשובה שלילית – עוד בטרם שקלנו לעומק

הם כבר יודעים הכל ואמורים לקבל את כל ההחלטות על דעת עצמם בלבד.

אבל עובדה היא כי פעמים רבות אנו "עובדים על אוטומט" - ממחרים לקבל החלטה לכאן או לכאן ומיד משגרים אותה לאוויר העולם, מבלי "לשון על זה" כפי שנהוג לומר.

זה יכול להוביל לחיכוכים מיותרים עם הסובבים ועם מנהלינו, ובשל כך קבלת החלטות שאויות.

אבל לטעמי זה אף גרוע יותר כאשר אנו מספקים תשובה שלילית "מהמותן" לעובדינו או לעובדים חסרי הניסיון – אשר עלולים להסיק מכך כי דעתם אינה חשובה לנו, לדכאם ולמנוע מהם את הניסיון להעלות את דעתם בעתיד.

במקרים אלו אנו עלולים לצאת בהפסד מר, כי דווקא אלו אשר עדיין אינם תפוסים בהטיות יכולים לעיתים להביא רעיונות שוברי מוסכמות, ובמיוחד עובדים חדשים אשר עדיין לא התרגלו למגבלות המערכת – יכולים לזהות את הפגעים שאנו כבר עיוורים להם.

לעיתים כדאי לנסות ולהעלות הרעיון על הנייר, לציין יתרונות וחסרונות – מתוך הבנה כי עצם החשיבה המסודרת הנדרשת להעלאת הדברים ונימוקם בכתב, יאפשר לנו להתעמק בהם יותר, ולעיתים לקבל תובנות מעמיקות וברורות יותר ומהן החלטות איכותיות ויעילות יותר.

פעמים רבות אנו "עובדים על אוטומט" ממחרים לקבל החלטה לכאן או לכאן

אז בהחלט ייתכן כי ניסיונונו מהעבר והרגלי העבודה שפיתחנו הם הנכונים, אך באותה מידה ייתכן כי הגיע הזמן לסטות מן ההרגלים ומההטיות שסובבות אותנו, ולכן המלצתי היא – **אל תמהרו לתת תשובה, במקרים רבים יהיה נכון בהרבה לומר "אני אחשוב על כך"** ולחזור אחרי יום-יומיים עם תשובה שקולה. רק אל תשכחו לחזור! עם תשובה.

חומר קריאה נוסף

<http://www.mkltesthead.com/2013/08/listen-then-form-your-own-opinion-99.html>

נשמח לשמוע רעיונות הערות והארות מכם הקוראים – בפורום בקב' הפייסבוק.

אתם מוזמנים לקרוא טיפים רבים נוספים ב: <http://goo.gl/UcW42>

ולחוסף טיפים משלכם לרשימה משותפת זו.

סדרת טיפים זו "כיצד להפוך לבודקים טובים יותר" מתבססת על דיון ב: Software Testing Club

[99 Things Testers Can Do To Become Better Testers](#)

ה-eBook החינמי שנוצר בעקבות דיון זה: [99ThingsEbook.pdf](#)

וסדרת פוסטים מאת [Michael Larsen](#) בשם: [Ways Workshop 99](#) - בה מיכאל מרחיב על כל אייטם וגם מספק הנחיות כיצד לתרגל הנושא.





תקציר המפגשים של קהילת הבודקים וקישור לסיכומים ומצגות.
מפגשים אלו מאפשרים שיתוף ידע, ביסוס וחיזוק היכולות המקצועיות בקרב הקהילה.

המפגש של

חודש מרץ התארח בחברת [StartHub](#) והתמקד בסביבת הענן, ביכולות ובכלים הזמינים בה. במפגש היו שלוש הרצאות:
הרצאה של [אביחי חג'י](#) מ-[cloudage](#). אביחי הציג בפני הקהל את תשתיות הענן שמספקות ע"י Microsoft Azure ומאפשרות הקמה וניהול נוחים של סביבות מורכבות לצורכי בדיקות או לצורכי פרישת מערכת מבצעת מלאה.

[מצגת](#)

הרצאה של [דמיטרי ינובסקי](#) מחברת פרפקטו מובייל שסיפר על היתרונות שמספקת פרפקטו בכל הקשור לבדיקות מכשירים ניידים, על הנוחות שמספק ענן של מכשירים מנוהלים ועל תשתיות המעטפת שפרפקטו מספקת כדי להקל על אוטומציה של מגוון מכשירים.

[מצגת](#)

[אורן רובין](#) מ-[Testim.io](#) סיפר על השימוש בבינה מלאכותית כדי לזהות אלמנטים בדפי אינטרנט ובכך להתמודד עם אחת הבעיות המציקות ביותר לכל מי שכותב אוטומציה - שינוי עיצוב ששובר את המבדקים האוטומטיים בלי סיבה "אמיתית". אורן סיפר על כך שטסטים, כשירות ענן, מסוגל לספק ללקוחותיו גם הקלטות של הבדיקות שרצו ומקום בו ניתן לשמור את הבדיקות ההיסטוריות.

[מצגת](#)

כפי שניתן לראות כל אחת מההרצאות סיפקה מידע שחשוב להכיר, ושלוש זוויות מבט על הענן - אחת של ספק תשתיות ענן, אחת של פלטפורמה שמהווה מעין ענן ייעודי למכשירים ניידים ואחת של שירות שנסמך על שירותי ענן כדי להציע ללקוחותיו מוצר עם לא מעט ערך מוסף.

המפגש של חודש מאי התארח בחברת [WalkMe](#) והתמקד בניהול זמן.

[מיכאל קריבוס](#) פתח והסביר לקהל שאי אפשר לנהל זמן, אבל אפשר לנהל משימות. הוא המליץ על הצגה חזותית של המשימות העתידיות כדי לאפשר לנו להתנהל בצורה מושכלת ועל הקדשת זמן בתחילת כל פרק זמן כדי להגדיר את המשימות איתן נתמודד בפרק זמן זה - עשר דקות בתחילת כל יום כדי לסדר משימות למשך היום, שעתיים-שלוש בתחילת כל שבוע כדי להגדיר את המשימות שנרצה לטפל בהן במהלך אותו שבוע, וכך הלאה. הוא גם הציג לקהל את לוח השנה שלו, בו הוא מסמן כל תאריך יעד, ואת זמן ההכנה הנדרש כדי להגיע אליו מוכן.

[יוניט טרבליס אלבז](#) הציגה את הדרך המעשית למימוש ניהול לוחות הזמנים באמצעות ניהול עצמי. יוניט נתנה דגש להבדלים בין דחוף לחשוב, הדרך לבניית מסלול מדויק לתיעוד המשימות ובכך האירה את השביל לאפקטיביות ויעילות תוך איתור וביטול גולני הזמן החיצוניים והפנימיים.

יוניט, קירבה את הנושא הכללי של ניהול זמן אל עולם הבדיקות כשסיפרה על שני צוותים שניהלה והגיעו להספק עבודה דומה למרות שצוות אחד עבד פחות שעות ביום וההבדל המרכזי לא היה ביכולות המקצועיות של הבודקים, אלא דווקא בכך שבצוות אחד לא הייתה תרבות של "מריחה" או של הפרעות חוזרות ונשנות שהצוות מייצר לעצמו (רמז - הצוות הממוקד והאפקטיבי לא היה ישראלי).

[מצגת](#)



TestIL - מפגשים של בודקים למען הבודקים לקידום מפגשי בודקים - אנו מחפשים

מקום!

מעוניין לארח מפגש בודקים?
הזדמנות להכיר בודקים אחרים ולעודד בקבוצת התעניינות בבדיקות!

מציאים!

יש לך רעיון או נושא שתצטרך לחלוק ולהציג?
הזדמנות מצוינת להצטרף ולהתפתח
בקבוצת המרצים!



test.il@yahoo.com

052-6718757

ניצן



לקבלת עידכונים שוטפים והרשמה למפגשי המיטאפ
היכנסו [לקישור](#)

המפגש של

חודש יוני התארח בחברת RSA בהרצאה הראשונה הסביר [איליה טרמן](#) מ-RSA על מהות הנדסת ביצועים (engineering Performance) ועל החשיבות שיש בתפיסת ביצועים של כל הצוות. הוא סיפר על ארבעה פרמטרים חשובים של ביצועים: מהירות, קיבולת (היכולת להכיל את נפח התעבורה הדרוש), יכולת גדילה (Scalability) ואמינות (Reliability). איליה חלק מניסיונו העשיר בתחום בדיקות הביצועים והזכיר את הדבר החשוב ביותר - יש המון מה ללמוד בתחום, אבל לא חייבים לדעת הכל כדי להתחיל.

מצגת

את ההרצאה השנייה העביר [עמית ורטהיימר](#) מ-RSA שדיבר על כלים בעולם הבדיקות. כבודקי תוכנה אנחנו משתמשים בחיי היומיום בלא מעט כלים - מדפדפנים ועורכי מסמכים ועד לסביבות פיתוח וכלי עומסים. כל אחד מהכלים האלה משנה את הדרך בה אנחנו חושבים ואת הבדיקות שאנו מסוגלים לבצע (או חושבים שצריך לבצע). עמית תיאר את האופן בו כלים משנים את דרכי החשיבה שלנו (מקלים על משימות, מספקים יכולות חדשות, מרכזים ידע וחושפים אספקטים של התוכנה שלא היו נראים לעין בלעדיהם). עמית המליץ לשים לב לכלים בהם אנחנו משתמשים, וחלק עם הקהל כמה טכניקות למציאת כלים חדשים - כי ככל שחגורת הכלים של בודק רחבה יותר, כך ניתן לבצע יותר משימות באפקטיביות רבה יותר.

מצגת

מה אתם חושבים?

נתי אתם מרגישים סיפוק מצבאתכם כבודקים?

הצבעה לגיליון 14



http://bit.ly/TW14_Poll



דעתכם חשובה, אנו מזמינים אתכם להשתתף בכל גיליון בהצבעה אשר את תוצאותיה נשתף עמכם בגיליון הבא



אתר המותאם למחשב בנייד. אומנם היינו צריכים לבצע גלילות אין סופיות לצדדים בכדי לראות את כל התפריט אך פונקציונאליות זה היה אפשרי. שהגיעו האתרים המותאמים הם נבנו בצורה אידיאלית לניידים גם בהתחשב בכך שהלחיצות נעשות על הכפתורים על ידי אצבע (רחבה לעיתים..) ולא על ידי חוד חץ העכבר. שהגיעו האתרים התגובתיים נולד ויכוח מי מהם מספק חווית משתמש טובה יותר. התשובה לרוב היא כי האתרים המותאמים מספקים חווית משתמש טובה יותר מהאתרים התגובתיים בשל בנייתם בקוד (כולל שטחי תגובה של כפתורים) עבור טלפונים ניידים אבל ההבדלים המזעריים יחסית בחוויית המשתמש לא מצדיקים תחזוקת שני אתרים ולכן המגמה היא לעבור לאתר תגובתי.

תפריט של אתר תגובתי בתצוגת מחשב:



תפריט של אתר תגובתי בתצוגת טלפון נייד:



התפריט זו דוגמה קלאסית של שינוי עיצוב אתר בשינוי גודל התצוגה. כאשר בתצוגת המחשב התפריט כולו פרוש לאורך החלון ובתצוגת הטלפון הנייד הוא מתקבץ לתוך תפריט לחיצ.

שיטת העיצוב והטכנולוגיה של אתר תגובתי

כיום מי שרוצה לבנות אתר, סביר להניח שירצה לבנות אותו כאתר תגובתי. אין צורך להמציא את הגלגל, ניתן למצוא ברשת שפע של תבניות לאתרים תגובתיים. התבניות הנפוצות ביותר הם של [Bootstrap](#) ואפילו ניתן למצוא תבנית דוגמה להורדה ב-[W3Schools](#) אחרי שמורידים תבנית מהאתר אתם מקבלים בעצם אתר תגובתי

כבר לא מעט שנים שהטלפונים החכמים הם חלק מחיינו. הם התחילו כסוג של גאדג'ט מעבר לטלפון הרגיל לצורך פנאי ותחביבים והפכו עם הזמן לכלי שימוש רב תחומי. בין אם זה בעבודה: דוא"ל, ניהול מידע משימות ועוד ובין אם זה בבית לרשימות טלפונים, קניות ומשימות אישיות. אני עוד זוכר את הפרסומות של חברת פלאפון לפני שנים של "יש לך Google בטלפון!" ומעכשיו אתה בעצם הופך את המכשיר הקטן שבידך לכלי רב עוצמה! כל המידע זמין בכף ידך במקום לחכות ש"תהיה ליד מחשב" כדי לבדוק משהו. גם את הקניות האינטרנטיות התחלנו לעשות בטלפון ועם אבולוציית הטכנולוגיה הגענו למצב שכשאנו בבית על הספה אנו לא קמים למחשב כדי לבצע פעולה כלשהי אלא שולפים את הטלפון החכם מהכיס. היום זה גם כולל עריכת מסמכים, שליחה להדפסה ופעולות שהיו שייכות רק למחשבים הנייחים.

עקב כך אתרי אינטרנט רבים (כמעט כולם) מתאימים את עצמם לטלפונים הניידים. כולנו מכירים שיש ל-y.net את m.y.net ול-b144 את m.b144. ובעצם כל האתרים יצרו אתרים כפולים. אחד עם תצוגה ויזואלית מותאמת למחשבים שולחניים והשני מותאם לטלפונים הניידים. בשרת יושב תנאי הבדוק את סוג המכשיר אשר ביקש את השאילתה לקבלת האתר, ושולח לו את האתר המותאם לסלולר (m.y.net לצורך דוגמה) או למחשב (y.net לצורך דוגמה).

עד כאן נשמע פשוט, זה אולי פשוט למשתמשי הקצה אך לא לעולם הפיתוח (שאנשי הבדיקות הם חלק ממנו). נוצר מצב המצריך תחזוקה של שני אתרים, בדיקת שני אתרים, לעשות רגרסיה לשני אתרים, לפתוח באגים לשני אתרים וכו'. בעיקרון עשו את זה לא מרצון אלא כי לא ממש הייתה ברירה. כל הנתונים הסטטיסטיים הראו את המגמה ההולכת וגוברת בשימוש האתר לטלפונים הניידים והירידה בשימוש באתרים המותאמים למחשבים. מה שכולנו כזכרנים חוונו כמובן ולכן בעלי האתרים היו חייבים לספק פתרון וייצרו עוד אתר שיותאם ויזואלית לטלפונים הניידים.

אני זוכר את הישיבות במחלקת שיווק כר"צ QA עם האנליסטים בהם מנחים אותנו כל פעם לאט לאט לשים יותר ויותר דגש בטלפונים הניידים ופחות במסכי המחשב עד אשר הגיעה הפעם שבה הציגו נתונים של 90% גלישה בניידים לעומת 10% במסכי מחשב. זאת לא הייתה הפתעה גדולה, כבר ב-2012 מכירות הטלפונים החכמים בארץ עברו את היקף מכירת המחשבים כך שהמידע כבר היה על השולחן אבל כשזה הגיע באופן רשמי זה הצריך לא מעט עבודה מצוותי QA-הן שכן רוב המשאבים היו מוכוונים סביבת מחשב (למשל אוטומציה מלאה לכל בדיקות הרגרסיה הייתה במחשבים לעומת הניידים שבוצע בהם אוטומציה לבדיקות שפיות ותהליכים מרכזיים בלבד).

ככל שהמגמה עברה לכיוון הטלפונים הניידים עבודת התחזוקה הכפולה גברה. שכן אתר מלא למחשב כבר יש. האתר המותאם לטלפונים הניידים שהתחיל כאתר עם חלק מהמידע והמאפיינים הקיימים במחשבים הלך וגדל עד אשר נהפך לאתר באותו סדר גודל. כדי לפתור את בעיית המשאבים העולמית הזאת נולד האתר התגובתי (Responsive).

מהו אתר תגובתי (Responsive)?

אתר תגובתי הינו אתר עם תבניות CSS (זו תבנית המגדירה חוקים עיצוביים לדף אינטרנט) שיוודעות להתנהג ויזואלית בצורות שונות בגדלי מסך שונים. כך שכאשר אפתח את האתר שלי במחשב אראה את האתר בצורה אחת וכאשר אמזער את החלון עוד ועוד עד שאגיע לרוזולוציה מסוימת (המוגדרת בתבניות) העיצוב ישתנה והאתר יוצג בצורה אחרת.

יווצר מצב שהגולש במחשב יקבל אתר מותאם למחשב ומי שגולש בטלפון יקבל אתר המותאם לטלפון, רק במקום שכל הקוד ישתנה כמו שנעשה עד היום באתרים המותאמים, הקוד יישאר אותו דבר ורק תבנית העיצוב תשתנה.

הנקודה של שינוי הרוזולוציה שבה קובץ ה-CSS מתחלף נקראת "נקודת שבירה" שכאשר מגיעים אליה התנהגות ה-CSS משתנה. ניתן להגדיר מספר נקודות שבירה בקוד במידה ויש חשיבות לכך אך נהוג לשבור את האתר ל-2 נקודות (ניידים ומחשב) או מקסימום ל-3 נקודות (ניידים, טאבלט ומחשב).

מה עדיף, אתר תגובתי או אתר מותאם?

כל יצירת האתרים המותאמים נולדו בשל בעיות עיצוב. הפונקציונאליות לרוב (מלבד פלשים וכדו') עבדה היטב גם שפתחנו



מוכן וכל שעליכם לעשות זה לשחק עם הגדרות ה-CSS כך שיתאימו לצרכים שלכם במדויק (מבחינת עיצוב כמובן).
כאשר מעצב גרפי מעצב אתר תגובתי עליו לעצב אותו בשיטת ה"גרידים" זאת אומרת שהוא מעצב את האתר בכמה טורים וכאשר אנו נקטין את הרזולוציה של האתר אלמנטים שהיו זה לצד זה יהפכו להיות זה מעל זה ויסתדרו בטור. כך תישאר רק גלילה למעלה ולמטה ולא גלילה לצדדים.



האתגרים של בדיקות תגובתיות

עם כל הרווחה שנוצרה במשרדי פיתוח התוכנה לאחר שהוקם האתר התגובתי אנשי ה-QA עדיין לא יכולים להצטרף לחגיגה במלואה כי הבדיקות עדיין צריכות להתבצע על שתי הפלטפורמות. אומנם הסיכויים לבאגים פונקציונליים להימצא רק באחת הפלטפורמות פחתו משמעותית אבל הם עדיין קיימים – כמו שמיד נראה, ועוד נשארה מרבית העבודה בבדיקות העיצוביות.

השאלה המרכזית היא איזה פינות ניתן לעגל כאשר נגשים לבדיקות של אתר תגובתי. אני יודע שלאנשי QA המילים "לעגל פינות" לא אמורות להופיע בלקסיון אבל הרבה פעמים אנו נמצאים בלחצי זמן לפני שחרור גרסה ואנו צריכים לבצע תיעוד של העבודה שיש לבצע, וחשוב לדעת היכן לסמוך על הסבירות שאם בדקתי פלטפורמה אחת אפשר לוותר על אותם הבדיקות בפלטפורמה השנייה והיכן לא ניתן.

בדיקות פונקציונליות

גם אתר תגובתי ככל הנראה יכיל שינויים פונקציונליים בין תצוגת המחשב לתצוגת הטלפונים הניידים. לדוגמה, בלא מעט אתרים לניידים מקובל להציג אייקון של טלפון אשר לחיצה עליו מעבירה לחיגן. במחשב כמובן שאין חיגן ולכן צריך לוודא שהאייקון נעלם או מקבל פונקציונליות אחרת. כנ"ל לגבי קישור המנווט לעסק מדף ה"צור קשר". באתרים לניידים רבים ניתן למצוא לצד המפה כפתור "נווט לעסק" אשר לחיצה עליו תעביר אותנו לאפליקציית Waze עם פרמטר של כתובת העסק. במחשב אין את אפליקציית Waze ולכן גם כאן צריך להחליט אם הכפתור נעלם או מקבל פונקציונליות אחרת כגון קישור לממשק מפות כזה או אחר. כמובן שרכיבים אלו חייבים להיבדק באופן מלא בשני הפלטפורמות ולא יכולים לרדת כשאנו מבצעים תיעוד של הנושאים. יתר הרכיבים הפונקציונליים, שההתנהגות שלהם זהה בשתי הפלטפורמות כגון לינקים, כפתורים, טפסים ועוד במידה ופלטפורמה אחת נבדקה בהצלחה הסבירות היא שגם בפלטפורמה השנייה לא ימצאו בעיות שכן בסופו של דבר מדובר באותו הקוד. קורה לפעמים ש-CSS פוגע בפונקציונליות, שטחי לחיצה של כפתורים וכו' יכולים להידרס על ידי שינוי ה-CSS אבל זה יותר נדיר ובהעדר זמן לבדיקות אפשר לרוב להסתפק בבדיקות פלטפורמה אחת בלבד. כל בודק מכיר מהן נקודות התורפה של המוצר שלו ויודע היכן צריך לשים דגש גם בזמן של לחץ זמנים.

בדיקות עיצוביות

בדיקות עיצוביות צריכות מאיתנו יותר תשומת לב. על אף שהתבניות הפופולריות כגון Bootstrap עושות עבודה טובה ולרוב המעברים די חלקים עדיין יש צורך לבנות מסמך בדיקות נפרד לבדיקות תצוגה

לניידים. למרות שהתבניות המוכנות עושות עבודה טובה כמעט ואין אתר אחד שלא ייאלץ להוסיף עליו כמות נכבדת של תנאים משלו (אם אתה ברזולוציה X ומטה תעשה ככה ואם אתה ברזולוציית X ומעלה תעשה אחרת) ולכן צריך לבדוק היטב את המצבים השונים.

את בדיקות המחשב מן הסתם עלינו לבצע דרך מחשב שולחני ואת בדיקות הטלפונים הניידים עדיף לבדוק ממכשיר נייד. אבל מעבר לכך באתר תגובתי נוספו לנו בדיקות שלא קיימות בבדיקות של אתרים אחרים ואלו בדיקות לניידים במחשב שולחני. כלומר, האם השינוי של קובץ ה-CSS מתרחש בבדיקת ברזולוציה שהוגדרה לו.

לא בכל מקום יש מגוון רחב של מכשירים לבדיקה, נניח והבודק לוקח מכשיר מסוג איפון 5 ושם הוא רואה שיש לו תצוגה לנייד זה לא אומר שבעלי איפון 6 שיש להם מסך גדול יותר גם יקבלו תצוגה של אתר לניידים ולא של מחשב. לכן כדאי לרשום לעצמנו את הרזולוציה שבה העיצוב אמור להתחלף ובמחשב השולחני למזער את החלון עד לרזולוציה המדויקת ולראות בדיוק באיזו נקודה העיצוב מתחלף.

איך ממזערים את החלון בדיוק לרזולוציה מסוימת? תוכלו לראות שיטה נוחה לזה בחלק של הטיפים בסוף הכתבה.

אוטומציה של אתרים תגובתיים

שאנו מדברים על אוטומציה של אתרים תגובתיים עולות אותן התהיות שעלו לנו בבדיקות הידניות, רק שבאוטומציה עניין תעדוף הבדיקות הוא לרוב קל יותר כי אפשר להריץ יותר בדיקות בזמן הנתון. מי שכתב אוטומציה על אתר המותאם למחשב בנפרד ואתר המותאם לניידים בנפרד ועכשיו עובר לאתר תגובתי יחוזה כמה הקלות משמעותיות בעבודתו. ההקלה המשמעותית היא שהיהיויים של המחשב והניידים הם אותם זיהויים!

(הדוגמה מטה מתייחסת לסלונים אך זה נכון גם לכלים אחרים).

אם אני לא משתמש ב-XPath (מה שכולנו נמנעים לעשות ככל הניתן) אז היהיויים שלי יוכלו לשמש אותי לשתי הפלטפורמות, מה שאומר תחזוקה קלה יותר משמעותית של הקוד (עוד סיבה טובה להשתמש ב-page object).

כמו כן ניתן לפתוח את דפדפן כרום במצב רגיל ופעם במצב סלולר (הקוד מצורף בסוף בחלק של הטיפים).

ופשוט לעשות את הבדיקות גם דרך האמולטור של דפדפן כרום – מבלי להתחיל להתעסק עם Appium.

לא נוכל לוותר על Appium כאשר אנו בודקים אפליקציית ניידים (אפליקציה שנכתבה באופן ייעודי למערכת הפעלה לניידים ולא אפליקציה שמבוססת על web) אבל כאשר בודקים אתר מותאם לניידים נוכל לכסות את רוב הפונקציונליות של האתר דרך האמולטור של כרום (תלוי כמובן בסוג האתר, באתרי תוכן וודאי שזה נכון). לא ניתן לבדוק דרכו פיצ'רים כגון מצלמה, חיגן טלפון, GPS ועוד אבל את הרוב המוחלט כן נוכל. כמו כן, בדומה לבדיקות הידניות המצריכות דגש על הבדיקות העיצוביות כאן גם נצטרך להשתמש במחלקות שבדקות CSS ולהשוות את התוצאות בין מצב למחשב למצב הנייד. יש אפשרות באוטומציה לשלוף את גדלי הגופנים, צבעים, גדלי אלמנטים ועוד. נוכל לקחת את הנתונים הללו ולהשוות אותם לשינויים שאנו מצפים שהתרחשו בהם בין הפלטפורמות השונות. וכן ניתן להשתמש בכלי השוואת תמונות כאלו ואחרות (כגון [sikuli](#) או [applitools eyes](#)) לצורך הבדיקות היוזאוליות.

טיפים לבדיקות אתר תגובתי

1. מומלץ לערוך רשימה של כל הרכיבים שמשתנים פונקציונלית בנקודות השבירה השונות (או לסמן אותם בצורה מסוימת בתוך המסמך בדיקות) ולתת להם עדיפות בבדיקות בכל אחת מהפלטפורמות.
2. מומלץ לערוך בדיקה במכשיר נייד בעל המסך הקטן ביותר הניתן על ידי הפיתוח, שם תמונות וכפתורים עלולים להשתבש.
3. הגדר מראש את סוגי וגדלי המכשירים הנבדקים, יש המון סוגים וגדלים ולא נוכל לבדוק את כולם. יש לגבש רשימה ממצה ואפשרית מבחינת הבדיקות "בדיקת ערכי קצה" אשר ישמשו כאסמכתא מספקת לכל סוגי וגדלי המכשירים.



```
System.setProperty("webdriver.chrome.driver", "C:\\**//  
chromedriver.exe");  
Map<String, String> mobileEmulation = new HashMap<String,  
String>();  
mobileEmulation.put("deviceName", "Google Nexus 5");  
Map<String, Object> chromeOptions = new HashMap<String,  
Object>();  
chromeOptions.put("mobileEmulation", mobileEmulation);  
DesiredCapabilities capabilities = DesiredCapabilities.  
chrome();  
capabilities.setCapability(ChromeOptions.CAPABILITY,  
chromeOptions);  
WebDriver driver = new ChromeDriver(capabilities)
```

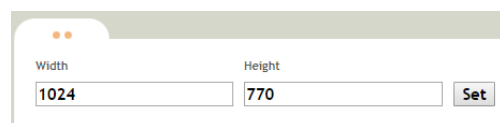
7. לקריאה נוספת מומלץ להיכנס לקישור הבא :

<http://blog.perfecto.io/responsive-web-design/responsive-web-design-testing-tips/>

4. לגבי אוטומציה תמיד מומלץ לעבור על הדפים גם בסקירת עין אנושית. בדיקות חזותיות מאתגרות יותר באוטומציה מאשר בדיקות פונקציונליות. בבדיקות פונקציונליות לרוב זה שחור ולבן. או שזה עובד או שלא. בבדיקות עיצוביות יכולות להיות סטיות לכל מיני כיוונים וכן דקויות שלא תמיד קל להכריע האם זה באג או שזה תקין.

5. יש לשים לב לטאבלטים. בעל המוצר צריך להחליט האם טאבלט מקבל תצוגת מחשב או נייד. מצד אחד הטאבלט הוא נייד ולעיתים אף קטן כמכשיר טלפון נייד אך עם זאת גם מתפקד כמחשב שכן גם אצלו לא ניתן להעביר על ידי לחיצה על האייקון לחיגון טלפון וכדו'.

כלי למזעור חלון לרזולוציה המדויקת, מאוד פרקטי. נכנסים לאתר הזה: <https://browsersize.com/>



כותבים את גודל הרזולוציה המבוקשת ולוחצים על set:

יפתח לנו חלון דפדפן בדיוק בגודל שהזנו עכשיו נוכל להכניס לחלון הזה את כתובת האתר שלנו ולראות אם אנחנו מקבלים את התצוגה המצופה.

6. קוד לפתיחת דפדפן כרום במצב של טלפונים ניידים (java, selenium webdriver) :



בחן את עצמך | טל פאר



הפתרון לשאלה

אלו מהמשפטים הבאים נכון לגבי בודקה (Testware)?

I. הבודקה קשורה לגרסה מסוימת של מערכת, ויש ליצור אותה מחדש עם כל גרסה חדשה

II. יש לנהל את כל הבודקה במערכת ניהול תצורה (Configuration Management)

III. בסוף הפרויקט יש להעביר את כל הבודקה לארגון שאחראי על תחזוקה

IV. המפתחים אחראים על התכנה נכונה של הבודקה

השאלה הזו מתייחסת לפרק 1.4.5 בסילבוס שמדבר על פעילויות סגירה הבדיקות, השלב הסופי בתהליך הבדיקות. זה נושא ברמת K1, כלומר זהו נושא שצריך לזכור.

בואו נעבור על התשובות האפשריות:

I – זוהי לא תשובה נכונה. ארגון בדיקות יעיל שומר על הבודקה שהקים ועושה שימוש חוזר ברכיבים המתאימים לגרסאות שונות כמו גם מפתח רכיבים חדשים שיתאימו לצרכי בדיקה ומטרת בדיקה חדשות, שעולות עם גרסאות חדשות של המערכת.

II – זוהי תשובה נכונה. שמירה על הבודקה במערכת ניהול תצורה תאפשר זיהוי של רכיבי מערכת הבדיקות, בקרת גרסאות, התאמה בין רכיבי מערכת הבדיקות לגרסאות המערכת הנבדקת, שמירה על נעקבות בין רכיבי הבדיקות השונים (תהליכי בדיקות, מקרי בדיקות, דוח פגמים וכו') לבסיס הבדיקות.

III – זוהי תשובה נכונה. בסיום הפרויקט יש לסגור את כל הבודקה ולהעביר אותה לארגון שאחראי על תחזוקה לצורך שימוש בעתיד. שימוש כזה יכול להיעשות, לדוגמה, כאשר מתקבל דו"ח תקלה מלקוח שמשתמש במערכת ויש לשחזר את התקלה על הגרסה הספציפית שבה משתמש הלקוח.

IV – זוהי תשובה לא נכונה. סביבת הבדיקות מוקמת, מותקנת ומנוהלת על ידי הבודקים שמכירים את כל רכיבי מערכת הבדיקות, את כלי הבדיקות ואת הגרסאות המתוקנות בה.

ניתן לסכם שתשובות II, ו-III הן נכונות ולכן התשובה הנכונה היא תשובה ב'.



THE STATE OF THE Software Testing Profession 2017-2018

Results from the Third Annual Survey



TECHWELL™

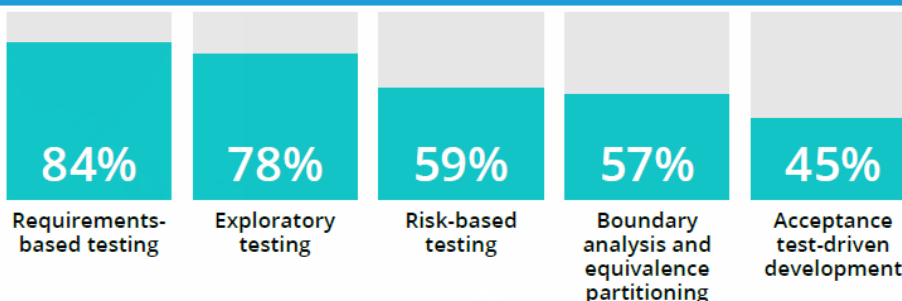
טור זה מפוזר במספר בלוקים ברחבי המגזין ומציג
ממצאי סקרים המציגים מגמות מרחבי העולם לעיונכם.
היום אנו מציגים חלק מתוצאות
[הסקר השנתי אודות מצב מקצוע בדיקות התוכנה 2017-2018](#)
שנערך ע"י Techwell.

יאן ברון



מנהל בדיקות בחברת
iMDSoft ישראל.
בעל תואר M.Sc במדעי
המחשב
והסמכה בינלאומית של
ISTQB FL
עם 35 שנות ניסיון
בהנדסת תוכנה וניהול
בדיקות.
תחומי עניין
מתודולוגיה וכלי פיתוח
לאזורים שונים בבדיקות,
בדיקות אוטומציה,
ניהול מחזור חיי מוצר:
מתודולוגיה וכלים.
התנדבות
ITSQB® – ראש ועדת
סקרים, ITCB®
ועד המנהלים,
SiGIST ישראל – יו"ר
תכנית הכנסים.

TOP 5 TESTING APPROACHES:



WHAT PRIMARY LIFECYCLE MODEL DOES YOUR TEAM USE?

Agile-Scrum	45%
Agile-Waterfall hybrid	20%
Waterfall	10%
Agile-Other	7%
Kanban	6%
V-model	3%
Other	3%
Scrum-But	2%
Iterative (e.g., RUP)	2%
None	2%

DO YOU WANT TO CONTINUE WITH A CAREER AS A TESTER OR TEST MANAGER FOR THE FORESEEABLE FUTURE?

Yes	75%
Undecided	16%
No	6%
Not applicable	3%

WHAT DO YOU SEE AS THE BIGGEST TREND FOR THE TESTING PROFESSION OVER THE NEXT FIVE YEARS?

(Respondents could select more than one answer)

The need for testers to become more technical	35%
The move from traditional methodologies to agile approaches	23%
The increasing importance of mobile and embedded systems	11%
The need for testers to become domain experts (have business knowledge)	10%
The increasing importance of the Internet of Things (IoT)	6%
The move to cloud computing	6%
The increasing importance of big data	5%
Other (please specify)	4%



אסתר צבר

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



ניצן גולדנברג

ניצן גולדנברג מזה 3 שנים בתחום בדיקות התוכנה, תפקיד נוכחי מהנדס בדיקות בחברת SeatGeek, המוביל הראשי של קבוצת המיטאפ TestIL



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.



דורון בר

מככב משנת 1998 ברשימת המבוקשים של הבאגים. הוא לחם בחזיתות רבות (חברות הזנק וחברות גלובליות) בדרגות שונות (בודק, ר"צ, מנהל ועוד). ביום דורון מנהל מתודולוגיות בדיקה בחברת Verint Systems ובליה בלוגר.



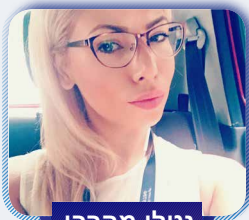
משה מאמיה

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



רון פרץ

הוא בודק אגיל, חוקר מערכות ומאמן עם למעלה מעשר שנות ניסיון בתעשיית התרופות. בזמנו הפנוי, רון מדריך ומתרגל אייקידו באינטגרל דוג'ו בת"א ולומד פילוסופיה, מתמטיקה ורשתות למידה עמוקה.



נטלי מהרבן

בת 28, נכון להיום מתגוררת במנילה שבפיליפינים. בעלת 9 שנות ניסיון וכיום יועצת בדיקות ארגוניות בחברת Truelogic ph. הקימה צוות בסטארטאפ האמריקאי whipclip במשך קרוב לשנתיים ולפני כן ניהלה את מחלקת הבדיקות בחברת קו מנחה - ביזפורטל. נטלי מתנדבת, רצה, טסה ושואפת למצוינות והתפתחות אישית.



טל פאר

בעל ניסיון של מעל 20 שנים כבודק ומנהל בדיקות במגוון חברות וטכנולוגיות במודלי פיתוח שונים. טל עובד כיועץ עצמאי ומדריך בחברה Practical Testing שהקים, כשהוא מלווה ארגונים בהקמת צוותי בדיקות, הטמעת מתודולוגית בדיקות, שיפור תהליכי בדיקה והדרכות בדיקות. טל הינו חבר ב-ITCB ומזכיר ארגון ISTQB.



עמית ורטהיימר

בודק תוכנה ב-RSA.

למה לכם לחשוב אם פספסתם???



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

bit.ly/TW-Reg



הרשמה

