

ביצוע בדיקות מתקדמות בסביבות טכנולוגיות חדשניות

פרופ' רון קנת, מוסד שמואל נאמן, טכניון KPA בע"מ

מיכל שלומי, יו"ר משותף המכון לניהול אג'ילי, מרצה במכללה האקדמית כנרת

חברי הנהלת איגוד מדעי הנתונים בלשכת המהנדסים

היכולת שלנו **ללמוד מטעויות** היא אבן יסוד בשיפור מתמיד, בכל תחום, בסביבות פיתוח ותפעול, כאשר התגובה הראשונית לטעות היא חיפוש אשמים, אנחנו חונקים את תהליך הלמידה ומונעים שיפור משמעותי. ד"ר דמינג הגדיר זאת בעקרון הניהול השמיני שלו שהתמקד בהסרת פחד בארגון, ראה

[/https://deming.org/explore/fourteen-points](https://deming.org/explore/fourteen-points)

בפועל אנו נתקלים בסימנים פיזיים ורגשיים המאותתים על טעות עוד לפני שהחשיבה הרציונלית נכנסת לפעולה. לכן, במקום להתמקד בשאלה "מי אשם", רצוי **לאמץ גישה המעודדת פסק זמן, התבוננות ולמידה מעמיקה** בכל טעות. השאלה החשובה יותר היא האם הטעות ראויה לתיקון, והאם כעת הוא הזמן המתאים לכך.

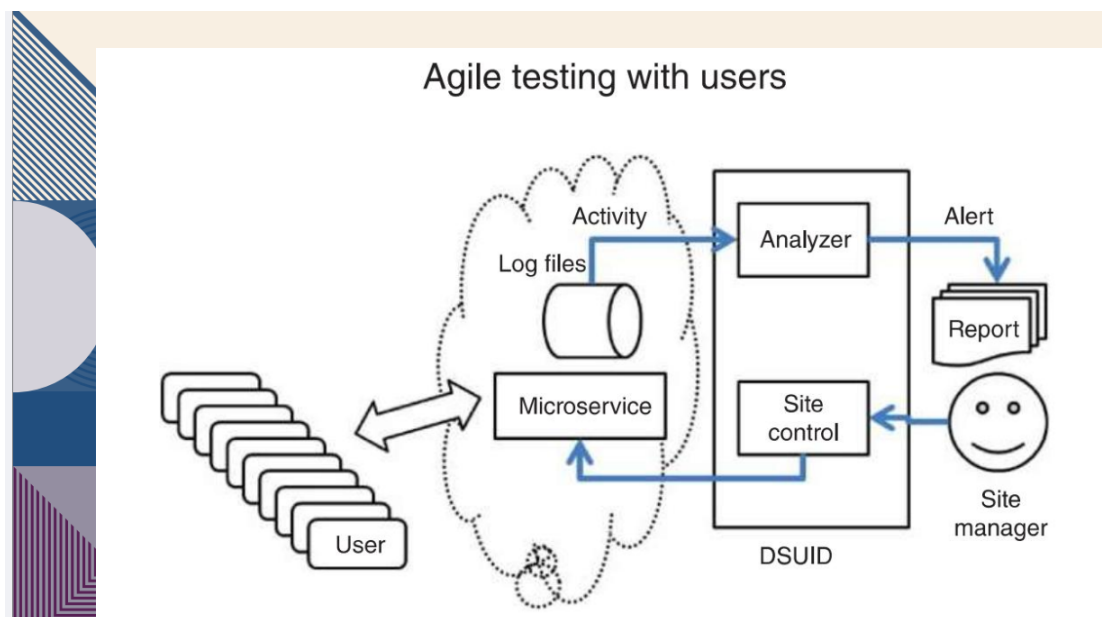
אחת הדרכים היעילות ביותר למנוע טעויות, מלכתחילה, או לזהותן בשלבים מוקדמים היא **שילוב תהליכי בקרת איכות** כחלק בלתי נפרד מתהליכי פיתוח מערכת או פרויקט. אימוץ גישה מבוססת על גישה פרואקטיבית לשיפור האיכות. תפיסה זו משמעותית במיוחד בסביבות מורכבות ודינמיות. נבחן לדוגמא, את העבודה של קבוצת פיתוח מערכת טכנולוגית שנועדה לאפשר התאוששות מאסון (disaster recovery), המיקוד בתרחיש היחוס הוא החיפוש אחר דרכים לגלות תקלות בסביבת הלקוח, עוד לפני שמשתמשים מודעים לתקלה או מדווחים עליה, כלומר **איתור יזום של תקלות** מהווה יעד מרכזי. בדוגמה זו, זיהוי צבירת נתונים בשרתי אחסון, עקב גידול בכמות הנתונים, מהווה נקודת אזהרה לזליגה במערכת הגיבויים החמה. כדי לשמור על רצף תפעולי יש לבצע פעולות במערך האחסון. בגישה כזו, המתמקדת בזיהוי סיכונים מוקדם, מתאפשר טיפול בתקלה עוד לפני התרחשותה. זה חוסך זמן ומשאבים ומעלה את שביעות הרצון של המשתמשים במערכת. הגישה הזו **מתקרבת לסביבת הייצור של הלקוח** וללמידה של מה קורה שם, על זה נרחיב ונעמיק.

בעידן של עדכוני תוכנה מתמשכים ובפיתוח דינמי, תהליכי בקרת האיכות הופכים למורכבים יותר. מהנדסי איכות אינם יכולים להסתמך רק על בדיקות מסורתיות, המבוססות על שלבים סדורים מראש של כתיבת בדיקות, הרצתן ותיקון תקלות. בסביבה המשתנה במהירות בה קצב הפיתוח המהיר, כמו בסביבת SaaS - Software as a Service, מחשוב ענן ומחשוב קצה, מתודולוגיות בדיקה מיושנות הן בלתי רלוונטיות. האתגר הגדול ביותר הוא להבטיח שהתוכנה תישאר יציבה ושימושית גם כאשר היא מתעדכנת כל הזמן. שינוי בקוד עלול לגרום לשגיאות חדשות או להוביל לתקלות בחווית המשתמש.

בדיקות אג'יליות משנות את הכללים! נפרוט את הסיסמא לכמה שיטות עבודה מיטביות: 💡

1. **מעקב מתמשך אחר נתוני משתמשים וזיהוי בעיות בזמן אמת** - במקום לבצע בדיקות בסוף מחזור הפיתוח, הבדיקות האג'יליות משלבות ניטור תמידי של פעילות המשתמשים בסביבת הייצור.
2. **אוטומציה של תהליכי בדיקות איכות לצמצום שגיאות אנוש** - ככל שהמערכות מורכבות יותר וקצב השינוי עולה, כך גדל הצורך באוטומציה מקיפה של תהליכי בדיקה, החל מבדיקות יחידה ועד בדיקות אינטגרציה ובדיקות מערכת.
3. **שימוש בבינה מלאכותית (AI) לחיזוי בעיות לפני שהן מתרחשות** - AI ולמידת מכונה מאפשרות לנתח דפוסים בנתוני שימוש ולזהות חריגות שעשויות להעיד על בעיות עתידיות.
4. **ניטור דינמי של השפעת עדכוני תוכנה על חוויית המשתמש** - הבדיקות מתבצעות בסביבת הייצור (Testing in Production - TiP) תוך שימוש בתעבורת משתמשים ולקוחות אמיתית, על ידי כך, הבדיקות מאפשרות לוודא את הפונקציונליות, הביצועים והערך של העדכונים אל מול נפח ומהירות התעבורה בפועל.

💡 לאסופת ההמלצות נציע מודל . DSUID Decision Support for User - Interface Design , מערכת חכמה לבדיקות איכות אוטומטיות. גישה חדשנית ומשלבת ניתוח נתוני משתמשים, אלגוריתמים מתקדמים ולמידת מכונה כדי לזהות תקלות ולשפר את הממשק עוד לפני שמשתמשים חווים בעיה.



מודל זה **משלב שימושיות ושיקולי חווית משתמש בתכנון**, בדיקה, פריסה ותפעול של מערכות שיתופיות דינמיות. הוא מאפשר זיהוי וביטול של אלמנטים מועדים לשגיאות בממשק המשתמש. הליבה של המתודולוגיה הזו היא אמצעי לניתוח מתמשך של פעילות המשתמש, המציע אמצעי תיקון להפחתת שיעור שגיאות המשתמש. המתודולוגיה משלבת בקרת תהליכים סטטיסטית של מדדי שירות תוכנה, המתקבלים על ידי מערכת תמיכה בהחלטות לתכנון ממשק משתמש, שבה המשתמשים הם אלמנטים, או שחקנים בלולאת הבקרה.

מה DSUID עושה בפועל?

 אוספת ומנתחת יומני פעילות משתמשים - לדוגמא כמות אחסון בשרתים אל מול הקיבולת

 מזהה דפוסי שגיאות חוזרים וליקויי שימושיות - לדוגמא ניתוקים בזרימת המידע אל שרת הגיבוי. יומני פעילות המשתמשים מספקים בדרך כלל חותמות זמן לכל פעולות המשתמש. בנוסף, כאשר קובצי יומן שרת זמינים, הם כוללים גם עקבות של קבצי תמונה וסקריפטים המשמשים לתצוגת המסך. חותמות הזמן של הקבצים הנוספים מאפשרות לנו להעריך שלושה מרווחי זמן חשובים: זמן ההמתנה של המשתמשים עד לתחילת הורדת הקובץ (מדידת היענות המסך), זמן ההורדה (מדידת ביצועי המסך), והזמן מהשלמת ההורדה ועד לבקשת המשתמש למסך הבא (זמן קריאת תוכן המסך).

 מזהירה מראש על אזורים בעייתיים במערכת - לדוגמא שליחת התראות למשתמשי המפתח

מאפשרת למהנדסי איכות לקבל החלטות מושכלות ולשפר ביצועים בזמן אמת

DSUID מספקת דיווחים אבחוניים מצרפיים המספקים משוב חיוני למתכנני ומנהלי השירות. אלה מתקבלים באמצעות ניתוח סטטיסטי מיוחד של רישומי פעילות המשתמשים המאוחסנים בקבצי יומן בשרתים. המודל של הבנת פעילויות המשתמשים מסייע לחזות את התגובות הסבירות ביותר של המשתמש למצבים חריגים.

במודל נעשה שימוש גם **במדדי בעיות שימושיות (- Usability Problem Indicators - UPIs)**. אלו הם אירועי משתמש המעידים על קושי פוטנציאלי של המשתמש. UPIs טיפוסיים הם פעולות משתמש ההופכות פעולות קודמות, כגון ניווט אחורה, ביטול, בחירה מתפריטים ראשיים או בקשות עזרה. שימוש ב-DSUID והטמעת בדיקות אגיליות מאפשרים לארגונים להפחית משמעותית את כמות התקלות במערכות שלהם, לשפר חוויית משתמש ולהתייעל מבחינת זמנים ומשאבים.

מדד נוסף הוא **מדד שירות התוכנה (PSI) וסיכוני שימושיות**

מדד שירות התוכנה (Program Service Index - PSI) הוא ציון שימושיות עבור מערכת או כל אחד ממרכיביה שאנו רוצים לשלוט בו. ה-PSI מתייחס למידת שביעות הרצון של הלקוח. הגישה המקובלת לבקרת שירות היא על ידי שימוש במושג "רמת שירות". ה-PSI מוגדר בשלב התכנון על ידי מתכנני השירות, וניתן להתאימו בשלב התפעול על ידי מנהלי השירות.

עליו לשקף את הסיכונים של שירות לקוי, ולכן עליו למדוד שני היבטים עיקריים של חוויית משתמש שלילית: ביצועי השירות ואמינות השירות. ניתן להגדיר PSIs נפרדים לביצועי שירות ואמינות שירות, או מדד משותף, למשל על ידי שילוב מדדים נפרדים. לרוב, הסכם רמת שירות נקבע כאחוזון, למשל 95%. משמעות הדבר היא, למשל, שב-95% מהזמן ה-PSI לא יעלה על מגבלה שנקבעה מראש.

כאשר מוצע שירות, תכונותיו כוללות מאפיינים שונים כגון ביצועים, אמינות, אבטחה ורמות שימושיות. שימושיות היא מונח המשמש לציון הקלות והאמינות שבהם אנשים יכולים להשתמש במערכת, כלי או אובייקט אחר מעשה ידי אדם על מנת להשיג מטרה מסוימת.

כשלים בשימושיות המיוחסים לטעויות אנוש הם בדרך כלל גורם מרכזי לכשל מערכות. לדוגמה, 60%-80 מתאונות המטוסים מיוחסות לטעות אנוש. כשלים בשימושיות אינם תמיד מזוהים נכונה. לדוגמה, שיחות רבות לתמיכה טכנית בנוגע לתקלות במערכות טלוויזיה מתבררות כתוצאה מטעות משתמש. ניהול סיכונים תפעולי מתמקד בזיהוי, סיווג והערכה של סיכוני כשלים תפעוליים המשפיעים על מערכות ועל בעלי העניין שלהן.

יישומי SaaS פופולריים מאפשרים למשתמשים לבחור מתוך רשימת שירותים ולשלב אותם באופן המתאים למטרת המשתמש. ההנחה היא שמחפשי השירותים פועלים לפי נהלים תפעוליים מסוימים כדי לתאר איזה שירות הם צריכים, ואינם עושים טעויות במפרט הדרישות. עם זאת, גמישות זו מציגה חששות שימושיות מיוחדים ומספקת יותר מדי הזדמנויות למשתמשים להיכשל על ידי הגדרת רצף שגוי או התעלמות ממגבלות מצב. לכן, יישומים מבוססי SaaS מועדים במיוחד לשגיאות, וה-PSI העיקרי המתאים ביותר לארכיטקטורות SaaS הוא אמינות תפעולית.

נסכם כי שימוש ב-DSUID סרמות שונות והטמעת בדיקות אג'ליות מאפשרים לארגונים להפחית משמעותית את כמות התקלות במערכות שלהם, לשפר את חוויית המשתמשים ולהתייעל מבחינת זמנים ומשאבים.

מה המשמעות עבור מהנדסי האיכות?

המהנדסים של העתיד צריכים לשלוט לא רק במתודולוגיות בדיקות מסורתיות, אלא גם בטכנולוגיות מתקדמות לניהול איכות, ניתוח נתונים ובינה מלאכותית. עליהם ליזום תהליכים המתקרבים לסביבת הלקוח, מנטרים אותה ומסיקים תובנות בזמן אמת כדי לייעל את התהליך ולהגדיל את אמינות הביצועים. מהנדסי איכות שרוצים להישאר בחזית החדשנות חייבים להבין כיצד לשלב ניתוח נתונים, AI ושיטות אג'ליות בתהליכי הבדיקה שלהם.

שימוש בגישות חדשות כמו DSUID מאפשר:

- ◀ זיהוי מוקדם של כשלים אפשריים במערכת
- ◀ שיפור תהליכי קבלת החלטות בבדיקות
- ◀ צמצום תקלות המגיעות למשתמש הקצה
- ◀ הפיכת בדיקות האיכות לחלק בלתי נפרד מהפיתוח – ולא שלב נפרד שמגיע מאוחר מדי

איך אתם מתמודדים עם אתגרי הבדיקות בסביבות דינמיות?

שתפו אותנו בתגובות :

האם כבר שילבתם בדיקות אג'ליות בארגון שלכם?

אילו כלים עוזרים לכם להבטיח יציבות ואיכות בסביבה משתנה?

📢 נשמח לשמוע את דעתכם ולפתח דיון מקצועי בנושא!

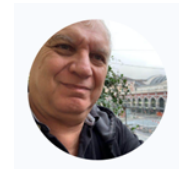
תגובות אפשר לכתוב בלוח [כאן](#)

למתעניינים נשמח לעניין אתכם להרשם לסדנה - **איכות 4.0: שילוב AI בתהליכי שיפור ארגוניים** כאן

הכתבה מתבססת על מאמר מקצועי מעמיק, הסוקר כיצד ניתן לשפר בדיקות אג'ליות באמצעות ניתוח נתוני משתמשים, בינה מלאכותית וכלים מתקדמים.

[לקריאת המאמר המלא](#)

Kenett, R. S., Harel, A., & Ruggeri, F. (2018). Agile Testing with User Data in Cloud and Edge Computing Environments. In R. S. Kenett, F. Ruggeri, & F. W. Faltin (Eds.), Analytic Methods in Systems and Software Testing (pp. 353–371). John Wiley & Sons Ltd



פרופ' רון קנת, יו"ר קבוצת KPA, פרופ' מחקר באוניברסיטת טורינו באיטליה ועמית מחקר בכיר במוסד שמואל נאמן בטכניון. חיבר 18 ספרים בנושאים של מדעי נתונים, סטטיסטיקה של מערכות בריאות, סטטיסטיקה תעשייתית וניהול סיכונים. יו"ר איגוד מדעי הנתונים, בלשכת המהנדסים, האדריכלים והאקדמאים במקצועות הטכנולוגיים בישראל.



מיכל שלומי - יו"ר משותף במכון לניהול אג'ילי, מומחית לניהול תהליכים ארגוניים, תרבות אג'ילית, אסטרטגיה, אפקטיביות ארגונית ואימון מנהלים. מאמנת אג'יל, יועצת לניהול תהליכים ומרצה באקדמיה. בעלת ניסיון רב בניהול פרויקטים וניהול מוצר. מהנדסת תעשייה וניהול, מוסמכת PMP, מנחת קבוצות בארגונים.

פרופ' רון קנת ומיכל שלומי