



מיכאל שטאל

הוא ארכיטקט בדיקות תוכנה באינטל, ישראל. במשך 17 השנים האחרונות מיכאל בדק מוצרים בתחום Wi-Fi, טלויזיות 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 כיפי).

מיכאל מציג תכופות בכנסים בארץ ובחו"ל והציג בין היתר ב-SIGIST Israel, STAR, QA&Test conferences ובכנסים אחרים. מיכאל מלמד בדיקות תוכנה בפקולטה למדעי המחשב באוניברסיטה העברית. ניתן לראות חלק מהמצגות והמאמרים שלו באתר www.testprincipia.com.

כפול או שניים?

בתקופה שבה התכונתי לבחינות הכניסה במתמטיקה לאוניברסיטה (כן, אנחנו מדברים על פרה-היסטוריה) נקראתי לחודש שמירות במילואים. כרגיל היה די הרבה זמן פנוי, כך שיכולתי להשקיע בלימודים. בצירוף מקרים נדיר אחד מהחיילים ששירתו איתי היה עולה חדש מרוסיה, פרופסור למתמטיקה. במשך חודש קיבלתי מורה אקדמאי צמוד שעזר לי להיזכר בחמש היחידות שלמדתי בתיכון.

מתי באג נחשב "כפול"? תלוי כמובן את מי שואלים



בשלב מסוים שאלתי אותו על מה הוא קיבל את הדוקטורט שלו. האיש, שלא דיבר עברית טוב, ענה באנגלית ובמבטא רוסי כבד: "They gave me a problem to solve".

...? שאלתי.

"It is not a problem anymore" הייתה התשובה.

גם בתחום שלנו ישנם דברים רבים שנפתרו כבר ויש הלכה סדורה איך לטפל בהם. דוגמה אחת היא ניהול באגים.

כיוון שבכל תוכנה יש באגים (לטובת ה-Job security שלנו), הרי שלכל ארגון יש תהליך איך מנהלים באגים. לעיתים זה דרך פגישה; במקרים אחרים דרך מיילים או בדיון ישיר עם מנהל הפיתוח. יהיה התהליך אשר יהיה - הוא מן הסתם קיים ומתנהל כבר זמן רב ועל פרויקטים שונים.

לכן תמיד מפתיע אותי מחדש איך זה שקיימים עדיין כמה אספקטים שיש עליהם וויכוח ולמרות שמזמן כבר היו צריכים להיכנס לקטגוריה של "not a problem anymore" הם עדיין לא סגורים עד הסוף. אספקט אחד הוא מתי באג נחשב כפול, כלומר, מתי הדיון שעומד לדיון הוא בעצם מיותר, כי יש כבר דיווח במערכת על אותו באג, והדיווח הנוכחי הוא שכפול של הקודם. במקרה כזה אפשר לסגור את הדיווח הנוכחי, אין צורך להשקיע בו זמן או לדון בו.

מתי באג נחשב "כפול"? תלוי כמובן את מי שואלים. על פי המפתחים, אם שני באגים נגרמים בשל אותה טעות בקוד הרי שמספיק לדווח על אחד, כי השני יתוקן ממילא ברגע שהראשון יתוקן. מבחינת המפתחים - והרבה פעמים גם מצד הנהלת הפרויקט - הטענה שמדובר בשני באגים שונים מערפלת את המידע שיש לנו על מצב איכות התוכנה.

בודקים צריכים לדווח "מה עשינו ומה קרה", ואילו "למה זה קרה" לא רלוונטי לדיון



הנה דוגמה משכנעת (לפחות מנהלי מוצר): נניח שיש לנו בסיס נתונים של ציוני סטודנטים. הקוד שולף נתונים מבסיס הנתונים על פי תעודת זהות אבל התוכנה מאפשרת למשתמש לשלוף נתונים גם על ידי נתונים אחרים; למשל: שם ושם משפחה; קוד סודי; מספר טלפון; מספר כרטיס אשראי. מהלך שליפת הנתונים עובד כך: בשלב ראשון משתמשים בפיסת המידע המזהה על מנת לשלוף את תעודת הזהות מטבלת פרטי הסטודנט. אחר כך מעבירים את מספר תעודת הזהות לקוד ששולף את הציון מטבלה אחרת (לכל מומחי בסיסי נתונים שמזדעקים שלא ככה זה עובד: תרגיעו... אמרתי "נניח" ©).

בשלב בדיקת המערכת הוגדרו חמש בדיקות, כל אחת מוודאת שליפה נכונה של הציונים כאשר מסופק פריט זיהוי אחר. עכשיו... מה יקרה אם יש בקוד באג שגורם לכך שבהינתן מספר תעודת זהות מסוים, נשלפים הנתונים הלא נכונים? כל חמש הבדיקות יכשלו!

אם כך, איך נתאר את איכות הקוד? האם נכון לדווח על חמישה באגים או רק באג אחד? אם נדווח על חמישה באגים הרי שנעביר מסר שמודול התוכנה ששולף נתוני ציונים הוא ברגרסיה נוראית וצריך להשקיע מאמץ ניכר לתקן אותו. אבל זה ממש לא נכון! יש רק טעות אחת בקוד וכל הבעיה תיפתר בחצי יום של מפתח - כולל ההרצה של Pre-checkin tests.

אז יש טענה למה לסגור ארבעה מתוך חמשת הבאגים כ"כפולים". אבל יש כמובן גם טענת נגד...

עד כמה אנחנו יכולים להיות בטוחים שהסיבה לכל חמשת הבאגים היא אותה סיבה? אכן, בדוגמה שלנו זה נראה די הגיוני, אבל בינינו - אם הם דפקו את הקוד של שליפת הציונים, זה לא רומז שהם יכולים גם לדפוק את הקוד של שליפת תעודת הזהות מהטבלה הראשונה? יש לי אמון רב במפתחים; אני בטוח שהם יכולים להכניס שתי טעויות שונות ל-release אחד! אם נסגור ארבעה מתוך חמשת הבאגים ככפולים, הרי שרק מכניזם אחד שגורם לבאג יתוקן. הטעויות האחרות ימשיכו להתקיים בקוד עד שנרץ את הבדיקות האלה שוב. אם אלה בדיקות בקדימות נמוכה, יתכן אפילו שהמוצר ישתחרר עם הבאגים האלה, ללא החלטה מודעת שזה מה שקורה.

מעבר לסיכון שהבאגים בסופו של דבר לא נגרמים מאותה טעות בקוד, סגירת באג כ"כפול" מעלה את הסיכון שלא נבצע עבורו בדיקות אישור (confirmation test) כשהבאג יתוקן. בדוגמה שהבאתי יש חשש שלא נבדוק בשליפת נתונים עובדת עם כל חמשת סוגי נתוני הזיהוי - אלא רק עם הנתון שתואר בבאג שנשאר פתוח ולא נסגר ככפול.

עדיף להיות סקפטי וברור לפני שמחליטים לסווג באגים ככפולים





הגישה שלי היא שבודקים צריכים לדווח "מה עשינו ומה קרה", ואילו "למה זה קרה" (שורש הבעיה - root cause) לא רלוונטי לדיון. הטענה של המפתחים שמדובר בכפילות נשענת על כך ש"מה קרה" וכנראה גם "למה זה קרה", זהים. טענת הנגד של הבודקים נשענת על כך ש"מה עשינו" - שונה. דעתי היא שבאג יכול להחשב ככפול אך ורק אם גם הפעולה שגרמה לו, וגם התוצאה השגויה, זהים.

עוד רמז מתי באג אינו כפול: בדרך כלל באג כפול נגרם בגלל ששני בודקים ראו את אותה הבעיה בדיוק, ושניהם דיווחו עליה בלי לבדוק קודם אם הבאג כבר נכתב במערכת. אפשר אם כך לומר בבטחון רב למדי שאם שני באגים שחשודים ככפולים דווחו על ידי אותו בודק, הסיכוי שהם דיווח כפול שואף לאפס. למה שאותו בודק ידווח פעמיים על אותו באג? זה סתם תוספת עבודה עבור הבודק שוודאי מודע לכך שכבר דיווח על התקלה. כנראה שהמופע של הבאג, גם אם נראה "אותו דבר" למפתחים, אינו בדיוק אותו דבר.

חשוב לנהל דיון עקרוני על הנושא ולהגיע להסכמות בין כל המשתתפים בניהול באגים מתי באג הוא כפול. קודם כל, זה ימנע וויכוחים חוזרים. אבל מדאיגה הרבה יותר היא ההשפעה שיש ל"יד קלה" בסיווג באגים ככפולים על הגישה של הבודקים לבאגים חדשים. אני חוויתי שתי תופעות:

- בודקים מוותרים על דיווח באג כי "ממילא יגידו לי שזה כפול"
- בודקים לוקחים את הגישה רחוק עוד יותר, וברגע שנראה להם שהם מבינים את ההשפעה של באג מסוים, נמנעים מלפתוח באג למרות שהוא ממש שונה מהראשון, בטענה שהסיבה לבאג השני היא אותה סיבה שגרמה לבאג הראשון.

המסקנה שלי היא שעדיף להיות סקפטי ובררן לפני שמחליטים לסווג באגים ככפולים. הנזק ממספר באגים כפולים במערכת נמוך מהנזק של מסר שצריך לצמצם את מספר דיווחי הבאגים. עדיף לטעות לצד המגזים; כלומר, ברגע שיש סיכוי שזה לא ממש אותה פעולה ואותה תוצאה - עדיף לשמור על שני הבאגים פתוחים. אם יתברר שתיקון אחד טיפל בשניהם, אז המפתחים זכו ב"אחד במחיר של שניים". מה שכן, צריך להקפיד לקשר בין הבאגים במערכת ניהול הבאגים. מעבר ליצירת link (שנתמך בכל מערכות ניהול באגים הנפוצות) כדאי לרשום במפורש בשני הדוחות: "יתכן והסיבה לבאג הנוכחי היא אותה סיבה שגרמת לבאג נוסף המקושר לדוח זה".

לא נותר לי אלא לקוות שסוף סוף הנושא הספציפי הזה יסווג גם הוא כ-not a problem anymore....

מישהו יכול להסביר?



"מגזין עולם הבדיקות" מחפש כותבים בנושאים הבאים



- ★ טריקים ובדיקות מהירות (Mnemonics, Heuristics)
- ★ כלים נוחים לשימוש (לתחומים כגון: Emb / Mobile / Security וכו')
- ★ השפעות של הפרעות קוגניטיביות בבדיקות
- ★ בדיקות מבוססות סיכונים
- ★ בדיקות קומבינטוריות