

מאמר זה התפרסם לראשונה במגזין "עולם הבדיקות" (גיליון 2, יולי 2015)

http://itcb.org.il/index.php?option=com_k2&view=item&id=754-מגזין-עולם-הבדיקות-גיליון-754&Itemid=865

פחות זה יותר

לפני שנים רבות, כשעבדתי בבדיקות ציפיים במפעל היצור של אינטל בירושלים (Fab8), הייתי שותף לוויכוח אופייני בין מפתחי מערכת אוטומציה. המפעל בירושלים לא עבד בשבת, והמטרה של מערכת האוטומציה הייתה לאפשר הרצת מכונות הבדיקה לתוך השבת בלי צורך במפעיל אנושי. הוויכוח היה על שפת התיכנות: C או Pascal? (כן... עברו כמה שנים מאז).

הטענות של שני הצדדים היו משמעותיות: מה קל יותר לעשות בכל שפה; עד כמה Strong Typing הוא משמעותי; איזו שפה יעילה יותר; באיזו שפה יש ספריות טובות יותר; וכו' וכו'. בנוסף, שני הצדדים טענו שאין ממש צורך להסכים על שפה אחת כי אפשר לכתוב מודולים בשפה אחת ולהשתמש בהם כספריות בשפה האחרת.

הוויכוחים נמשכו לאורך זמן, ואם זכרוני אינו מטעה אותי בסופו של דבר הקוד נכתב ברובו ב-C כי זאת הייתה השפה המועדפת על מי שכתב את רוב הקוד...

שנים אחר כך הנחיתי קבוצת בודקים בכתיבת מערכת לאוטומציה של בדיקות תוכנה – וחוו מהעובדה שהוויכוח שם היה על Visual Basic מול C#, השאר היה מאוד דומה: הוויכוחים האין סופיים; פגישות טלפוניות ופנים-אל-פנים במטרה לשבת עד שיצא עשן לבן; הבאת הוכחות מכל צד למה הגישה שלו נכונה - והכל תוך המשך עבודה במקביל בשתי השפות כשכל צד מנסה לייצר מספיק קוד על מנת להבטיח שהשפה "שלו" תבחר.

מי צודק?

לדעתי – כל אחד ואף אחד. הטיעונים של כל צד נכונים, וברוב המיקרים אי אפשר להכריע את הוויכוח על סמך טענות טכנולוגיות. השוואה של ה"בעד" של שפה אחת מול ה"בעד" של השפה השניה היא בעצם וויכוח על למה שתי השפות קיימות, וברור שלכל אחת יש יתרונות מסויימים. כשמשווים את היתרונות מול החסרונות של שפה אחת קשה להעריך איזה פרמטר חשוב יותר. זאת גם הסיבה שהוויכוחים מתמשכים: אין טיעון מחץ שמטה את הכף. למעשה, אי אפשר להגיע להסכמה המבוססת על עובדות והגיון. כל מהנדס ימשיך לחפש את הטיעון האולטימטיבי לטובת השפה "שלו".

אכן, יש מקרים שבהם הפתרון קל כי יש יתרון ברור וחד משמעי לאחת השפות בהקשר של הפרוייקט הנדון. אבל מה עושים כשזה לא המצב, ולכל שפה יתרונות וחסרונות?

במצב כזה חשוב להבין שהשיקול העיקרי צריך להיות **מספר השפות ולא איזו שפה**:

- (1) יש "אנרגיית אקטיבציה" לכל שפה נוספת. בכל מעבר משפה אחת לשניה המהנדסים יבזבזו זמן להזכר מחדש בפרטים טכניים. יתכן ויהיה צורך בשכפול כלים (כגון: IDE, Code Control) ומערכות תומכות כמו Continuous Integration - i Nightly build. המהנדסים יצטרכו ללמוד לשלוט ביותר כלים. כל זה עולה כסף וזמן.
- (2) מהנדסים לא אוהבים במיוחד לעשות שימוש חוזר בקוד של מישהו אחר. יש תמיד הרגשה של "ייקח לי פחות זמן לכתוב את זה בעצמי מחדש". הסיכוי לשימוש חוזר בקוד שאינו בשפת התכנות שהמהנדס בקיא בה, הוא מיזערי.
- (3) אם תצטרכו להעביר אחריות ממהנדס אחד לשני (עקב עזיבה, שינויי תפקיד וכו'), יתברר לכם שהשפה בה כתוב הקוד מצמצמת את מרחב התמרון שלכם מבחינת מועמדים מתאימים. גם בגיוס, תצטרכו לצמצם את החיפוש לאנשים ששולטים בצורה טובה בשתי שפות או לשלם את עקומת הלימוד של השפה שהמהנדס החדש לא בקיא בה.

מסקנה: ההחלטה הנדרשת היא **ניהולית** ולא **הנדסית**. על המנהל להקשיב לטיעונים משני הצדדים על מנת לוודא שאכן מדובר כאן במקרה של שוויון (אין שיקול ברור לאחד הכיוונים). ברגע שזה התברר, השיקולים הופכים ניהוליים ולא הנדסיים:

- באיזה כלי תמיכה הארגון משתמש?
- איזה שפה מוכרת היטב ביותר מהנדסים בקבוצה?
- מהי השפה המועדפת על האדם שעליו תוטל כתיבת רוב הקוד?
- מהי השפה שבה כתוב רוב הקוד שיש לכם כרגע?
- באיזה שפה כתובות מערכות שיתמשקו עם המערכת שלכם ויתכן שתצטרכו לעדכן אותן?
- וכו'

שימו לב שהשאלות לא דנות כלל בעדיפות הטכנולוגית של שפה אחת על השניה! בנוסף, אלה שאלות אובייקטיביות שקל לענות עליהן. אם מי מהן לא עוזרת להחלטה (למשל: אין שפה אחת שבאופן בולט ידועה יותר למהנדסים), אפשר להוריד את הפן הזה מסך השיקולים.

ומה אם אין אף תשובה שמכריעה את הכף? אז להשאר עם שתי שפות? במקרה כזה, גרירת ההחלטה לאורך זמן היא יקרה לא פחות מהעלות של בחירה קצת פחות נכונה. אין הבדל ממשי אם ההחלטה נופלת לכאן או לכאן. אז מכניסים יד לכיס ו...!

זורקים מטבע!