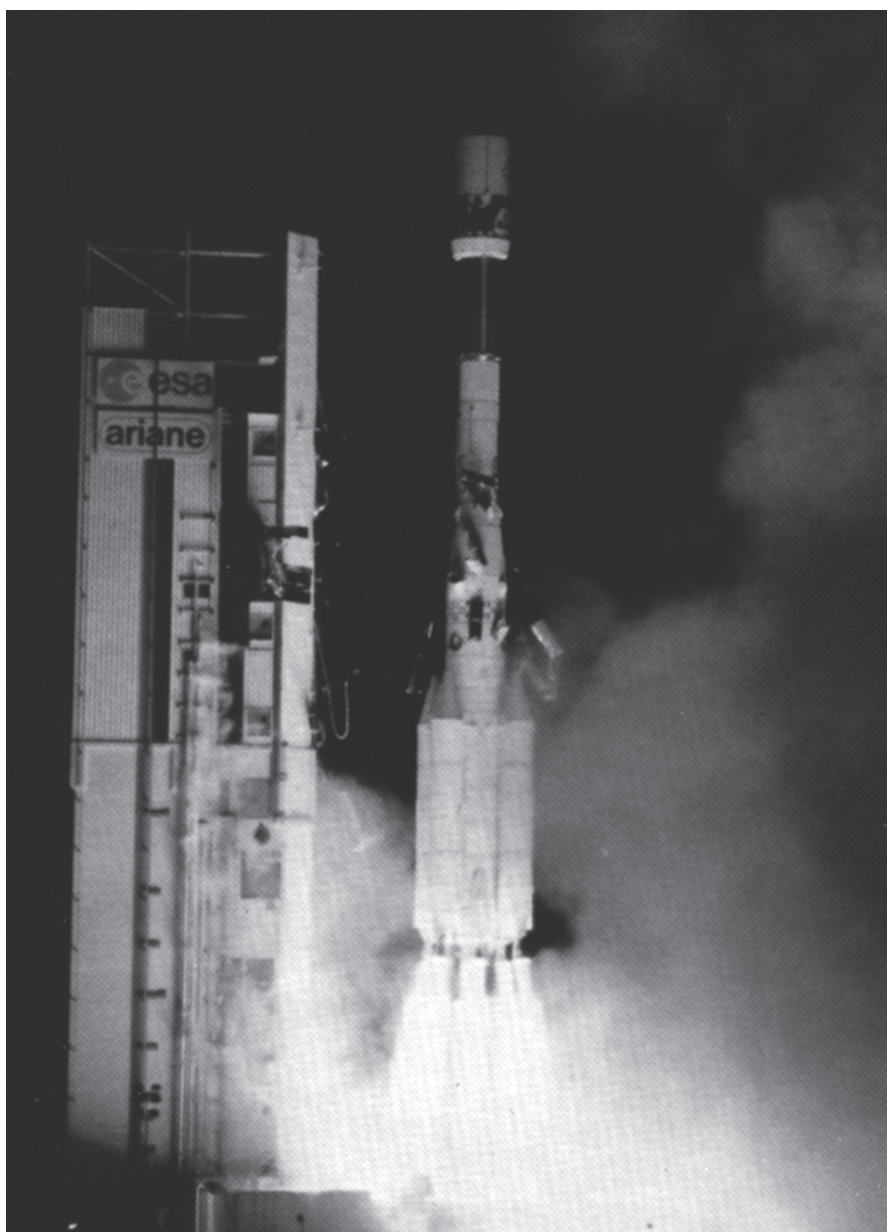


--

--



בדוק ונצמד?



ההכרה בחשיבות הביצוע של בדיקות מערכת ושל בדיקות תוכנה באמצעות גוף בלתי תלוי הולכת וגדלה עם התפתחות הטכנולוגיה ומורכבות המערכות בכלל והמערכות הצבאיות בפרט. כיצד מבטיחים כיסוי בדיקות נאות, ומה נדרש לעשות כאשר כישלון אינו אופציה?

אריאן 4 ממריא | אריאן 5 הוא ממשיכו של פרויקט אריאן 4. בפרויקט המשך נהוג - ונכון - להישען על מרב הנכסים מהפרויקט המקורי, אך יש לנקוט משנה זהירות כשהדבר מגיע לבדיקות



סא"ל (מיל) שלמה גור-אריה

סמנכ"ל בחברה העוסקת, בין השאר, בפיתוח מערכות עתירות תוכנה

מבוא

4:30 לפנות בוקר. ההכנות האחרונות נשלמות. קר באתר השיגור באלסקה, אך האווירה חמה ומתוחה. שר ההגנה בכבודו ובעצמו יעקוב אחר השיגור במעגל סגור מהפנטגון. השיגור חייב להצליח. שלוש שנות פיתוח מואץ יסתכמו בארבע דקות של מעוף אל מחוץ לאטמוספירה. קנקן הקפה שהתמלא הלילה בפעם השמינית, מהביל.

בחדר הבקרה מתנדנד על כיסאו קולונל ג'ונס. המתח ניכר בחיכוך. הוא עוד לא יודע ש-25 שניות לאחר השיגור ייכנס טיל הניסוי לתמרון לא מבוקר ויפול לאוקיינוס כאבן.

שעון הספירה לאחור רץ. הנשימה נעצרת. ציפורניים נכססות. שיגור. מנועי הטיל משגרים אותו בסערה השמימה ומותירים שובל עשן ארוך. וההמשך ידוע.

בכל זאת, חרף הכשל, המזל שיחק פעמיים: בראשונה מאחר שמדובר בניסוי ולא בשיגור מבצעי ובשנית, מאחר שהסיפור לא התרחש באמת... אבל סיפורים דומים שהתרחשו במציאות מוכרים היטב לכל מי שעוסק בפיתוח מערכות נשק בכלל ומערכות טילים בפרט.

כישלון ידוע מראש

מובן שלא כל אי-הצלחות בניסוי או בשיגור מקורן במיעוט בדיקות. לעיתים תכן גבולי, מזג אוויר גרוע או ביצוע מחוץ למעטפת המפרט הם שהופכים לרועץ. עם זאת מתועדים בספרות כשלים רבים שאותם ניתן לשייך לחוסר בדיקות תוכנה. מעניין במיוחד, ולכן ארחיב על כך, הוא נפילת הטיל אריאן 5. 37 שניות טיסה, שבהן הגיע הטיל לגובה של ארבעה ק"מ, חשפו למהנדסים טעות קטנה (והטעויות הן בדרך כלל קטנות) במערכת הניווט האירוציאלית. הטעות הזאת גרמה לטיל ענק הממדים והיקר לצאת מהמסלול, עד שלמנהלי הניסוי לא נותר אלא לחזות חפוי ראש בהשמדתו.

מהן הסיבות שמערכות מגיעות למשתמש כשהן לעיתים לא בדוקות במידה מספקת? כיצד מבטיחים כיסוי בדיקות נאות? כיצד ניתן לשפר את איכות הבדיקות ואת היקפן?

במאמר הזה אתמקד בבדיקות ובאמינות של תוכנה, אך מהן ניתן להשליך על בדיקות מערכת בכלל.

המשל של אריאן 5

תוכנית אריאן 5

תוכנית אריאן 5 של סוכנות החלל האירופית (ESA) נועדה להציב, בין השאר, לווייני תקשורת בחלל. מדובר בפרויקט הדגל של סוכנות החלל האירופית, והוא המשכה של תוכנית אריאן 4.

בפרויקט המשך נהוג - ונכון - להישען על מרב

הנכסים מהפרויקט המקורי, אך יש לנקוט משנה זהירות כשהדבר מגיע לבדיקות. יש לתכנן בקפידה את שלב הבדיקות בהתייחס ל"מה נשתנה", מה אין צורך לבדוק שוב, ומה נדרש לבדוק מההתחלה.

ב-4 ביוני 1996 שוגר הטיל הראשון בתוכנית אריאן 5. ערכו של הטיל ושל המטען שנשא נאמד בכחצי מיליארד דולר, אך מדובר היה ביותר מאשר הסכום הזה: השיגור סיכם 10 שנות פיתוח והשקעה של 7 מיליארד דולר. התכונה והמתח בחדר הבקרה היו בשיאם באותו בוקר גורלי.

מזג האוויר ביום השיגור באי קורו בגויאנה הצרפתית שעל החוף הצפוני של יבשת דרום-אמריקה היה נאה. לא נצפו רוחות חריגות, והסיכוי לסופות ברקים היה נמוך מאוד. הספירה לאחור הופסקה פעם אחת בשל ירידה בקריטריון הראות. בשעה 09:33 הוצתו מנועי הענק (שני מאיצים ומנוע ראש), והטיל החל לנסוק בסערה ובעוצמה. השיגור היה מושלם, והכול פעל כפי שתוכנן ונחזה בסימולציות. ככל שהשניות חלפו, והמשימה התקדמה באופן תקין, כך חזר אט-אט קצב הנשימה בחדר הבקרה לנורמלי. במשך 36 שניות ארוכות היה המעוף תקין לחלוטין.

"הכשל במשימה 501 של טיל אריאן 5 נגרם בשל אובדן מלא של נתוני ההנחיה. האובדן הזה נבע מטעויות באפיון ובתכן של תוכנת מערכת הניווט. בתוכנית הפיתוח לא נכללו אנליזות ובדיקות מספקות אשר היה בכוחן למנוע את הכשל"

התקלה

בשנייה ה-37 הכול החל להשתבש. בגובה 3.7 ק"מ החל הטיל לפתח תמרון בלתי מבוקר שגרר עומסים אווירודינמיים גדולים. חלקי טיל החלו להתפרק, ובכללם המנועים. המחשב המוטס המרכזי זיהה כי הטיל נכנס למצב טיסה בלתי מבוקר ולכן שלח אות השמדה עצמית. במקרה כזה מבצע הטיל בעצמו סדר פעולות אשר נועד להפסיק את המשימה תוך גרימת נזק מינימלי לסביבה. בעקבות הפיצוץ התפזרו חלקים זעירים מהאריאן 5 על פני שטח של 12 קמ"ר באזור הסוואנה של דרום אמריקה.

טילים כבדים הם איטיים בתחילת מעופם, ולכן אמצעי היגוי כמו כנפונים אינם יעילים. מהסיבה הזאת בראשית המעוף משמש בדרך כלל נחיר הפליטה של המנוע הרקטי לניהוג

(ניהוג וקטורלי). פקודות ההיגוי מגיעות ממחשב הניווט (מערכת ניווט אינרציאלית - מני"א) המודד כל הזמן את המיקום של הטיל ומורה לו כיצד ולכן להתכוון. המדידות והחישובים נעשים בתדירות גבוהה (כמה פעמים בשנייה) ומתורגמים לפקודות היגוי.

סדר האירועים כפי שנכתב בדו"ח החקירה היה כדלקמן:

- בשנייה ה-39 החלו מנועי הטיל להתפרק בשל תמרון חריף.
- התמרון החריף נבע מהטיה מקסימלית של נחירי המנועים.
- פקודת ההיגוי החריגה הגיעה מהתוכנה של מחשב הטיסה המוטס אשר קיבל נתונים שגויים ממערכת הניווט האירוציאלית.
- הנתונים השגויים נבעו מכך שמערכת הניווט קרסה בשל חריגה בתוכנה.
- הקריסה נבעה מהמרה שגויה ולא אפשרית בין שני מספרים מאופי שונה (ממספר ממשי למספר שלם).
- מחשב הגיבוי לא נכנס לפעולה מאחר שגם הוא שבק חיים בהיותו טעון בתוכנה זהה. הוא שבק עוד בטרם נכנסה לוגיקת הגיבוי לפעולה.

תחקור הכשל

בסיכום הדו"ח שכתבה ועדת חקירה שמינתה סוכנות החלל האירופית נקבע:

- תקלת תוכנה במחשב הניווט הזרימה נתונים שגויים למחשב הטיסה, וזה נתן לטיל פקודות היגוי קיצוניות.
- היגוי חריף גרם לטיל להתפרק - מה שהוביל לפקודת ההשמדה העצמית.
- התוכנה של מערכת הניווט האירוציאלית (מני"א) של אריאן 5 נלקחה ממערכת אריאן 4.
- התקלה הופיעה במודול (חבילת) תוכנה לאיפוס נתונים. המודול הזה מסיים את פעולתו באריאן 5 לפני השיגור, ואילו באריאן 4 ממשיך המודול הזה לפעול עד 40 שניות לאחר השיגור.
- ביום השיגור אובחנו רוחות אופקיות לא חזקות במיוחד. את הרוחות האלה צריך היה להכניס לתהליך איפוס הנתונים. מהירות אופקית גבוהה שנבעה מהרוחות בזמן השיגור הוכנסה לחישובים באופן שגוי על-פי הלוגיקה החדשה, ומכאן הייתה קצרה הדרך להתבררות.
- ועדת החקירה בראשות פרופ' ליאון (J.L. LION) קבעה באופן נחרץ: "הכשל במשימה 501 של טיל אריאן 5 נגרם בשל אובדן מלא של נתוני ההנחיה. האובדן הזה נבע מטעויות באפיון ובתכן של תוכנת מערכת הניווט. בתוכנית הפיתוח לא נכללו

אנליזות ובדיקות מספקות אשר היה בכוחן למנוע את הכשל".

האם אפשר היה אחרת?

פרויקט המשך הוא, כאמור, הליך מקובל ומוכר גם בישראל (למשל בטנק המרכבה). אולם העברת תוכנה או אלגוריתמיקה קיימת ממערכת אחת למערכת המשך מחייבת לבדוק לעומק מה השתנה, מה יש לבדוק מחדש ובאילו תנאים ומה לא. קשה לבדוק מערכות פירוטכניקה או הנעה רקטית מאחר שהן שקולות לעיתים לגפרורים שבהם בדיקה של האחד (יידלק או לא?) אינה מעידה על האחרים. מערכות כאלה מכונות One Shot Device. לעומת זאת במערכות מחשב או בחומרה משובצת מחשב ניתן לבדוק כמעט את כל קצות המערכת - בין היתר בסביבה מדומה הנקראת Hardware in the loop. להעדר בדיקות או לבדיקות בלתי מספקות יכולות להיות כמה סיבות:

1. הרצון להציב את הפרויקט בזמן (ובדיקות, כידוע, צורכות זמן).
 2. הרצון לחסוך משאבים (בדיקות צורכות כוח אדם וכסף).
 3. הערכה שגויה בנוגע להיקף הנדרש של הבדיקות ביחס למורכבות המערכת.
- תקלות נפוצות נוספות הן שילוב הבדיקות בשלב מאוחר מדי והעברת האחריות לבדיקות לגורם המפתח עצמו. נראה שכמה כשלים חברו יחדיו בתוכנית אריאן 5.
- בעקבות מסקנותיה של ועדת החקירה התעורר הצורך בבדיקות מסוג מסוים המכונות Coverage Testing. הבדיקות האלה מוודאות שמערכות שהן קריטיות למשימה עוברות כולן בקרה קפדנית. תקן הפיתוח למערכות תעופה - DOD178 - מחייב ומגדיר פורמלית את הבדיקות האלה.

הקו הדק שבין "כמעט נפגע" ל"נפגע" - או "טוב שעלינו על זה בזמן"

פעמים רבות במהלך שירותי הצבאי, עת נכחתי בניסויי ירי, בניסויי טיסה או בפעילות מבצעית כזאת או אחרת, עלתה במוחי השאלה אם המערכת שאנו כל כך רוצים שתצליח הצלחה מלאה בביצוע משימתה נבדקה במידה מספקת. להלן כמה דוגמאות לכמעט כשלים או תקלות שנבעו מליקויי תוכנה או מחוסר בבדיקות תוכנה.

- ספינת הטילים האמריקנית יורקטאון הייתה הראשונה בצי שצוידה במערכות אוטומציה ממוחשבות ומתקדמות שנועדו, בין השאר, לחסוך בכוח אדם. בספטמבר 1997 נדמו מנועי הספינה בעודה בלב ים למשך כשלוש שעות.

מסקנות החקירה העלו כי מערכת ההנעה שבקה חיים מאחר שמחשבי הספינה לא יכלו להתמודד עם חלוקה ב־0.0. כשאנו יושבים במשרד מול המחשב, והתוכנה מגיעה לשלב שבו היא נדרשת לבצע חלוקה ב־0, המחשב יתריע, והתוכנה תפסיק לעבוד. פחות סימפטי להיתקל במקרה כזה בספינת טילים בלב ים. חלוקה ב־0 ובעקבותיה התבררות של תוכנה היא כנראה מסוג הטעויות שלעולם חוזרות.

- במסגרת בדיקות מתקדמות של מכ"ם מסוים טרם מסירתו ללקוח התגלו מצבים שבהם המערכת נתקעת ומפסיקה לתפקד. הודות לערנותו של הבודק, אשר הורה לבצע את הבדיקות בכל מרחב המעטפת, נתגלה שבחישוב מסוים בתוכנת המכ"ם מתבצעת חלוקה בטווח האופקי. כאשר מדובר במטרה המופיעה בדיוק מלמעלה (טווח אופקי 0), התוכנה - ובעקבותיה המערכת כולה - נתקעת וקורסת.
- בבדיקות מקיפות שבוצעו בארץ לטיל שנרכש מחו"ל נתגלה באג בחישוב גזרות הירי. מדובר בטעות אלגוריתמית בסיסית: ניתוח דרממדי במקום ניתוח תלת־ממדי.
- בתחום הרפואה בולט אירוע שבו כמות האנרגיה שפולט מתקן רפואי קורן

בתחום הרפואה בולט אירוע שבו כמות האנרגיה שפולט מתקן רפואי קורן נשלטה באמצעות תוכנה שלא נבדקה נשלטה באמצעות תוכנה שלא נבדקה דיה. תקלת התוכנה - במכשיר Therac 25 - תוקנה לאחר שארבעה אנשים קיפחו את חייהם

נשלטה באמצעות תוכנה שלא נבדקה דיה. תקלת התוכנה - במכשיר Therac 25 - תוקנה לאחר שארבעה אנשים קיפחו את חייהם.

- תקלת תוכנה גרמה לכך שרישומים רפואיים (הכוללים המלצות לטיפולים ולתרופות) הוצמדו לאנשים הלא הנכונים. התקלה התגלתה בזמן הודות לבדיקות שנעשו לפני השימוש בתוכנה. אף שהתקלה הייתה "רק" במערכת מנהלית, הרי אילולא התגלתה בזמן, עלולים היו להיגרם נזקים כבדים עד כדי קיפוח חיי אדם.
- בתחום היבשה ישנן דוגמאות רבות של תקלות במערכות שונות שנתגלו בזמן וכאלה שנתגלו לאחר שנמסרו למשתמשים וללוחמים. למשל,

בבדיקות של טנק עם סיום פיתוחו וערב מסירתו למערך הלוחם התגלו כמה תקלות, אשר אילולא התגלו במועד עלולות היו לפגום מהותית ביכולתו המבצעית בשדה הקרב. למשל בכונת הטנק, שהיא מערכת עתירת תוכנה, התגלתה התקלה הבאה: באיפוס הכונת נמצא היסט בין הפקודה בתוכנה למיקום התותח בפועל בערכים שהיו גורמים לפגיעה הרחק מנקודת המכוון; חישוב אלגוריתם ההתחשבות ברוח היה הפוך, כך שהתיקון לרוח היה גורם להחטאה מהותית. ערנות ומודעות בתהליך הבדיקה הביאו לשדה מערכת אמינה ובדוקה.

- כיום אנו יודעים שיעילותה של מערכת טילי הפטריוט שהוצבה בסעודיה ערב מלחמת המפרץ הראשונה ירדה מ־96% ל־13% עקב תוכנה פגומה שגרמה לתקלה באיפוס השעון הפנימי.
- לעיתים כאשר אנו נתקלים בתקלת מחשב אנחנו פותרים את הבעיה באמצעות "טיפוח" על המסך, סינון קללה חרישית ולחיצה על כפתור ה־reset. במערכות גדולות וסבוכות ובמערכות המוגדרות "מערכות משימה קריטיות" - כמו מערכות צבאיות - תקלה כזאת היא בלתי נסלחת.

מסקנות

מה נדרש אפוא לעשות כדי להבטיח שהמערכת שאנו מפתחים, קולטים, מטמיעים או מפעילים היא כשירה ואמינה?

- בבדיקות מערכת בכלל ותוכנה בפרט הן מקצוע הדורש מומחיות, הכשרה וניסיון ייחודיים. את הבדיקות נכון שיתכננו ויבצעו אנשים שהוכשרו להיות בודקים. חברות בדיקה משקיעות רבות בהדרכה ובהכשרה של עובדיהן, לעיתים תוך בניית התמחות מגזרית ספציפית. (לבדיקת מערכות בתחום הצבאי, לדוגמה, נדרשים מומחיות וניסיון ייעודי שיכולים להיות שונים לגמרי מהמומחיות הנדרשת לביצוע בדיקות בתחום הפיננסי או הציבורי).
- מצויים בשוק כלי תכנון ובדיקות תוכנה ייחודיים שבכוחם גם לציין עד כמה התוכנה בדוקה. הבולטים שבהם: IBM רשיונאל, הכלים של חברת מרקורי, TRACS PRO ו־LDRA של חברת טאקט. נכון וראוי שהמפתחים יעשו שימוש בכלים הקיימים בשוק.
- על בודקי התוכנה להיות עצמאים ובלתי תלויים בגוף המפתח. כך מובטחת "עין בוחנת" אובייקטיבית. מקובל שהגוף המפתח או הלקוח שוכרים את שירותיה של חברת בדיקות כדי



אריאן 5 | 37 שניות טיסה, שבהן הגיע הטיל לגובה של ארבעה ק"מ, חשפו למהנדסים טעות קטנה - והטעויות הן בדרך כלל קטנות - במערכת הניווט האירוציאלית. הטעות הזאת גרמה לטיל ענק הממדים והיקר לצאת מהמסלול, עד שלמנהלי הניסוי לא נותר אלא לחזות חפויי ראש בהשמדתו

הנבדקת ישנם מאות מרכיבים שבכוחם להכשיל את המשימה כולה. אך בדבר אחד קולונל ג'ונס בטוח: הוא עשה כל שביכולתו והשתמש במרב האמצעים כדי להבטיח שהמערכת תהיה בדוקה ואמינה. את זה הוא חייב לממונים עליו. את זה הוא חייב לאזרחי מדינתו.

שעון הספירה לאחור רץ. הנשימה נעצרת. ציפורניים נכססות. שיגור. מנועי הטיל משגרים אותו בסערה השמימיה ומותירים שובל עשן ארוך... נותרו עוד 1, 2, 3, 4, 5 שניות לגמר המשימה. אנחת רווחה, צעקות רמות בחלל החדר. "עשינו זאת! ביצוע מושלם!" חיבוקים, והנשימה חוזרת לקצב נורמלי.

בהכנת הכתבה סייע מר ולדי נצר

על הלקוח או המשתמש המבצעי לוודא שהמפתח מבצע בדיקות בלתי תלויות באמצעות גוף שיעודו והתמחותו הם ביצוע בדיקות

לוחם בשטח או רופא בבית חולים. על הלקוח או המשתמש המבצעי לוודא שהמפתח מבצע בדיקות בלתי תלויות באמצעות גוף שיעודו והתמחותו הם ביצוע בדיקות.

4:30 לפנות בוקר. ההכנות האחרונות נשלמות. קר באתר השיגור באלסקה, אך האווירה חמה ומתוחה. שר ההגנה בכבודו ובעצמו יעקוב אחר השיגור במעגל סגור מהפנטגון. השיגור חייב להצליח. שלוש שנות פיתוח מואץ יסתכמו בארבע דקות של מעוף אל מחוץ לאטמוספירה. קנקן הקפה, שהתמלא הלילה בפעם השמינית, מהביל.

בחדר הבקרה מתנדנד על כיסאו קולונל ג'ונס. המתח ניכר בחיוכו. במערכת כה מורכבת כמו זו

שתבחן את תוצרי הפיתוח שתי וערב ובכל קצות המעטפת.

- יש להביא בחשבון את שלב הבדיקות כבר בשלב של תכנון הפרויקט, מאחר שהבדיקות צורכות עד כדי 50% מהמשאבים המוקצים לפיתוח התוכנה בפרויקטים צבאיים.
- יש לשלב את הבודקים בשלב מוקדם מספיק בפיתוח כך שייחסך זמן בלימוד התהליכים מחד גיסא, והצוות יהיה עדיין בלתי תלוי מאידך גיסא. הניסיון מלמד כי שילוב מוקדם חוסך זמן וכסף לארגון. היכרות אינטימית עם המערכת הנבדקת הופכת את הבודקים לעיתים למוקד ידע בהמשך חייו של הפרויקט.

סיכום

אמר מי שנחשב לאבי הנדסת התוכנה המודרנית גריידי בוך (Grady Booch): "הדבר האחרון שאני רוצה לשמוע בטיסה הוא את הקברניט שואל: 'האם יש איש תוכנה על המטוס?'"

הסוגיה של בדיקות מערכת אינה עניין רק למפתח אלא גם ללקוח או למשתמש הסופי - שיכול להיות