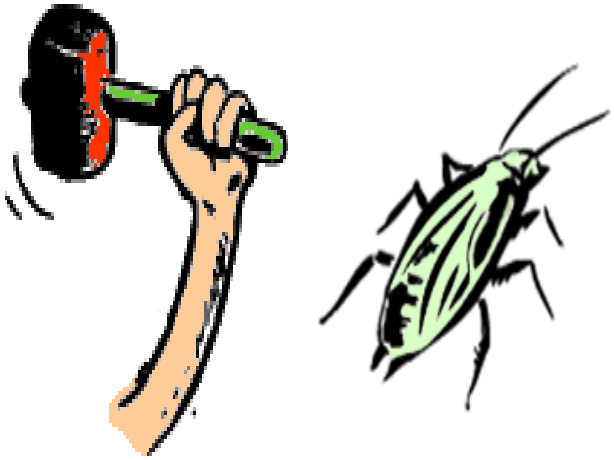
בדיקות: ריכוז מושגי יסוד



רשימת מושגים

- A/B Testing
- Acceptance Tests
- Alpha Testing
- Beta Testing
- Black Box Testing
- Branch testing
- Bug Life Cycle
- CAST
- Control Flow testing
- Debugging
- Equivalence Class
- Integration Testing
- Interface Testing
- Load Testing
- Monkey Tests
- Smoke Test
- Regression Testing
- Sanity Test
- Static Analyzer
- STD
- STP
- STR
- Stress Test
- Stub/Mockup
- System Tests
- TC- Test Case
- TDD
- Test
- Test Script
- Test Site
- Test suite
- Transaction flow testing
- Unit Testing
- Usability Testing
- Volume Testing
- White Box

A/B Testing

- בדיקת A/B (מבחן A/B) היא שיטה לבדיקת הקיים אל מול השינוי וקביעת הגרסה בעלת התוצאות החיוביות יותר.
- בדיקה זו נפוצה בתחומי השיווק, סחר אלקטרוני, פיתוח אתרים ותחומים נוספים ומאפשרת לבחון את תגובות המשתמשים לשינויים.
- הבדיקה מתבצעת על ידי השוואת 2 גרסאות, A- בדר"כ הגרסה הנוכחית (control) מול - B גרסה עם שינוי יחיד (treatment) שתי קבוצות משתמשים דומות מקבלות בו זמנית את המערכת הנבדקת, כל קבוצה גרסה אחרת. מודדים את מטרת הבדיקה ומחליטים איזו גרסה גרפה תוצאות טובות יותר.
- החלת שינוי לא מבטיחה שיפור ולכן יש סיכוי להפסד. ביצוע בדיקת A/Bמספק מידע אמין על שינויים לפני שמחילים אותם בפועל ותוצאות הבדיקה הן מקור מידע בעל ערך רב למקבלי ההחלטות. שימוש בבדיקה זו יוביל להפחתת סיכונים והוזלת עלויות.

Acceptance Tests

- מבחני קבלה אשר מבוצעים באחריות המשתמש-מכונות גם UAT
- דגש עיקרי: תאימות מירבית לתהליך הארגוני מנקודת המבט של המשתמש
- המטרה : לבדוק האם התוכנה בשלה להשתלב בפעילות העסקית של הלקוח
- הבדיקות מבוצעות בעיקר מול ה Use Cases והדרישות
- שיטת הבדיקה: קופסא שחורה
- מגמה מעניינת: Acceptance test-driven development: ATDD

Alpha Testing

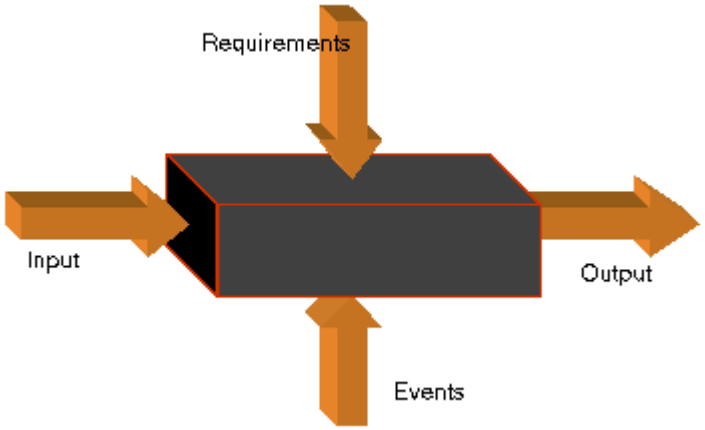
- מבדקים ראשוניים
- מבחני קבלה באחריות הלקוח
- המבחנים מבוצעים באתר הפיתוח בסביבה קרובה ככל האפשר לאתר הלקוח (אתר הייצור)
- גרסה לא רשמית

Beta Testing

- מבדקים שניים
- המטרה: בדיקת ישימות עקרונית
- בדיקות שנעשות על ידי לקוחות פוטנציאליים בסביבה שלהם ללא קשר לסביבת הפתוח
- מבדקים אלו מבוצעים בד"כ למוצרי מדף במטרה לקבל היזון חוזר מהשוק

Black Box

- אסטרטגיה לביצוע בדיקות
- שמות נרדפים:Behavioral testing, Functional testing
- בדיקות אלו מבוססות על השוואה מול דרישות
- תהליך הבדיקה מתעלם לחלוטין מהמבנה הפנימי של התוכנה



Branch Testing

- טכניקה אשר מטרתה להבטיח מעבר דרך כל מצבי התנאי

Testing technique to satisfy coverage criteria which require that for each decision point, each possible branch (outcome] be executed at least once

Bug Life Cycle

- New כאשר מזוהה לראשונה
- Open כאשר מחליטים שיש צורך לתקן
- Assigned כאשר יצאה הזמנת עבודה לתיקון
- Fixed כאשר הבג תוקן
- Closed וידוא הריגה
- Archived נשמר לצורך הפקת לקחים ושיפור נהלים

Cast

- Computer-Aided Software Testing
- כינוי לכלל התוכנות שמסיעות למכן את תהליך הבדיקות
- ריכוז מסודר של רשימת התוכנות ניתן למצוא ב

  http://www.testingfaqs.org/tools.htm
- תוכנות מובילות:
  - QC חברת HP (***Quality Center***)
  - MTM חברת מיקרוסופט (***Microsoft Test Manager***)

Control Flow Testing

- ביצוע בדיקות קופסא שחורה הנובע מניתוח המסלולים האפשריים
- ניתן להציג את הדרישות על ידי גרף מכוון ושימוש ב 3 מרכיבי זרימה: sequence , selection and iteration
- הגרפים משמשים לניתוח סטטי לצורך בנית ה TC

Debugging

- ניפוי

תהליך של ניתוח והסבר מדוייק
לשגיאה שנתגלתה בניסוי
כתוצאה מתהליך זה, יתוקן מרכיב התוכנה
והמרכיב יעבור ניסוי מחודש

Equivalence Class

- טווח / קבוצת ערכים של ערכי קלט/פלט אשר עבורם ההתנהגות של מרכיב התוכנה אמור להיות זהה על פי הדרישות
- דוגמא: במסך יש למלא את שדה הגובה לאדם בוגר.

  הטווח המותר הינו 70 ס"מ עד 230 ס"מ

  הערכים אותם מומלץ לבדוק: 69,70,71,100,229,230,231

Integration Testing

- בדיקת אינטראקציה ושלמות לוגית של יחידות שכבר עברו בדיקות
- הבדיקות מתמקדות בממשקים בין המרכיבים השונים
- הבדיקות מתבססות על מסמכי עיצוב ואינן מתימרות לזהות אי עמידה בדרישות הלוגיות

Interface Testing

- סדרת בדיקות אשר מטרתן לבדוק האם רכיבי המערכת מעבירים ביניהם מידע/פיקוח בצורה נכונה שעונה על הדרישות

Load Testing

- בדיקת העמסה מיועדת לבחון האם המערכת עומדת בעומס המשתמשים שבו היא נדרשת לעמוד. בדיקה זאת לא ניתן לבצע באופן ידני
- אחד הכלים המובילים בתחום הוא Load Runner של חברת HP (לשעבר Mercury)

Load Testing

DailyMaily גיליון 4653 יום ה', 27.3.2008
עורך אחראי: פלי הנמר | עורך ראשי: יהודה קונפורטס | עורך: אור יעקב | סגן עורך: מיכל ברונר

מועדון SQAT

"הנדסת ניהול העומסים היא התמחות בפני עצמה"

כך אמרו משתתפי מפגש מועדון SQAT של קבוצת ThePeople • ירון צוברי, דירקטור QA בקומברס: "בודקי עומסים הם אנשי מקצוע מבוקשים ביותר ומתוגמלים בהתאם" • שלומי עמר, מנכ"ל טסט פרו: "חייבים לבנות מודל עומסים בכל פרויקט תוכנה" • אלעד סנדר מ-V-Ness: "בדיקת עומסים היא חובה להצלחת פרויקט"

אבי בליזובסקי, מערכת DailyMaily, ThePeople

שלומי עמר, מנכ"ל טסט פרו

בדיקת תוכנה היתה במשך שנים בן חורג של ענף התוכנה, וכיום איש לא מערער על הצורך לבצע בדיקות כדי להיות בטוחים שהתוכנה תבצע את המוטל עליה. למרות זאת, עדיין מעט מאוד ארגונים מבצעים גם בדיקות עומסים כדי לוודא שביום פקודה - כאשר המערכת תחל לפעול - היא לא תקרוס. בנושא זה עסק המפגש השני שהתקיים השבוע, של מועדון SQAT המאגד את כל אנשי המקצוע בתחום בקרת איכות התוכנה.

שלומי עמר, מנכ"ל **טסט פרו**; **ואבי בטרן**, רכז ידע בטסט פרו, דיברו על הצורך לבנות בתחילת הפרויקט את מודל העומס (Load Model). "הפרמטרים שנצטרך לענות עליהם הם זמן תגובה - כיצד המערכת מגיבה לבקשות המשתמשים, והאם היא מספקת את התגובה הרצויה בזמן סביר; סקלביליות - האם המערכת תתמוך בעומס משתנה; יציבות - האם האפליקציה יציבה תחת עומס צפוי ולא צפוי של משתמשים - מצבי קצה; בטחון: האם אנו בטוחים שהמשתמשים יזכו לחוויה חיובית - השאלה החשובה שנצטרך לענות למנכ"ל". אמרו. לדבריהם, "את כל אחד מהפרמטרים הללו אנחנו רוצים לבחון ביחס לצפיות המשתמשים, מגבלות המערכת והעלויות".

עמר תיאר כיצד הצליח מודל העומסים לשפר ביצועים של מערכת שהוקמה במשרד האוצר, ואשר תפקידה היה לנסות לחלק בצורה הוגנת את התקציב שנועד להצלת קרנות הפנסיה בין העמיתים. "המערכת נועדה לכמה עשרות אנשים שעבדו כמזיני נתונים, אך בשל כך למעלה ממיליון גמלאים לא יכלו לקבל את הגמלה בזמן. הפתרון היה כמובן הלגדיל את מספר מזיני הנתונים, אך המערכת לא עמדה בעומס. במקום לקנות עוד שרתים ועוד קיבולת תקשורת, הצענו להם שינויים ארכיטקטוניים שחסכו להם קרוב למחצית מההוצאות שהיו צפויות להם לו היו בוחרים בפתרון הקל".

Monkey Tests

- בדיקות אקראיות ללא תיעוד וללא מתודולוגיה
- מבוצעות על ידי בודק בעל דמיון פרוע במטרה ל"הפיל אותה"

Regression Testing

- ביצוע חוזר לבדיקות שכבר בוצעו בעבר על מנת לוודא שתיקונים שהוכנסו לתוכנה לא גורמים לשגיאות בפונקציונליות שכבר עובדת
- אחת התוכנות המובילות שמסיעת לבצע את התהליך באופן אוטומטי היא WinRunner של HP

Sanity Testing

- להבטיח שהמערכת המועברת לבודקים אכן נמצאת במצב "בדיקתי"
- המטרה היא למנוע מצב בו מעבירים את המערכת לסביבת הבדיקות אבל המערכת עדיין "בוסר" ומתעופפת בכל נגיעה במקלדת
- רמה זו של בדיקות אינה כוללת בדיקת נכונות אלא רק בדיקת כשרות לבדיקות מקיפות

Smoke Test

- בדיקה שטחית לחלוטין של מרכיב תוכנה במטרה לראות שהוא מגיב ומראה סימנים שהוא עובד
- המינוח לקוח מתחום האלקטרוניקה , לאחר תיקון המכשיר מחברים אותו לחשמל ואם אין ריח עשן כנראה שהמכשיר תקין

Static Analyzer

- כלים שמסיעים להעריך את איכות התוכנה על ידי ניתוח פקודות המקור
- למשפחה זו שיכים כלים כגון: כלים ליצירה אוטומטית של תרשימי זרימה, כלים לבדיקת תקנים, כלים לקביעת סביכות ועוד

STD

- Software Test Details/Description
- תכנון וכתיבת תרחישי הבדיקות
- בכל תרחיש יש להציג מספר צעדים כאשר עבור כל צעד צריך להציג מהם נתוני הקלט ומהי התוצאה הצפויה

STP

- Software Test Planning
- הכנת תוכנית מסגרת לבדיקות
- התוכנית כוללת בין השאר:
  - הגדרת מדיניות בדיקות(Testing policy)
  - הגדרת מטרות הבדיקות(Testing Objectives)
  - הגדרת רמות הבדיקה הנדרשות
  - אילוצי לו"ז ומשאבים
  - הגדרת תנאי סף(Exit Criteria)
- תקנים: נוהל מפתח, IEEE829

STR

- Summary Test Reporting
- מסמך המסכם את מצב תוצאות הבדיקות.
- מבוצע בד"כ על ידי מנהל הבדיקות

Stress Test

- בדיקות שנועדו לבדוק את תגובות המערכת למאמץ מעבר לגבולות שהוצבו בדרישות
- הרעיון הוא להביא את המערכת למיצוי המשאבים (זיכרון פנימי, זיכרון חיצוני, CPU וכו') ועל ידי כך להביא את המערכת לנקודת שבירה
- הבדיקה תתמקד בסיכונים/נזקים שעלולים להיגרם בנקודות השבירה

Stub/Mockup

- קטע תוכנה מלאכותי שנועד לבצע סימולציה של התנהגות מרכיב תוכנה אחר שטרם נכתב
- הרעיון המרכזי הוא להיכנס לשלב בדיקות של יחידות תוכנה גם כאשר לא כולן כתובות עדיין

System Tests

- המטרה היא לבדוק את המערכת כולה במיוחד בהיבט הטכני
- בדיקות אלו מכונות גם בדיקות מסירה
- הנחת היסוד היא : כל המרכיבים עברו מבדקים קודמים בהצלחה
  Unit Testing, Integration Testing
- הבדיקות הן מסוג קופסא שחורה/אפורה ומתבססות על מסמך הדרישות

TC- Test Case

- תאור מפורט של בדיקה שיש לבצע כחלק מהבדיקות שמכסות UC נתון או מסלול ספציפי
- התיאור כולל: זיהוי, תנאים מוקדמים לביצוע(מצב המערכת),קלטים,תאור סדרת הפעולות שיש לבצע (Test Script) ו התוצאות הצפויות
- מקרי הבדיקה צריכים להיות מתועדים היטב ובנויים לשימוש רב פעמי

TDD

- Test Driven Development
- זוהי שיטת פיתוח המבוססת על עקרונות הבסיסיים כדלקמן:
  - לכל מרכיב תוכנה יש לתחזק במקביל מרכיב unit test שבודק אותו
  - אין להעביר קוד לביצוע אלא אם כן יש לו בדיקות כחלק אינטגרלי
  - בזמן פיתוח כותבים קודם את תוכנת הבדיקות
  - תוכנת הבדיקות קובעת בעצם מה צריך לכתוב

Test

תהליך של הפעלת מרכיב תוכנה
(מודול,מחלקה, תוכנית, מערכת ,תת מערכת)
במטרה לאתר מקסימום שגיאות/תקלות

Test basis

מסמך הבסיס ממנו נגזרו מקרי הבדיקה
מומלץ לנהל גרסאות ולאפשר השוואה ביניהן

Test Script

- תיעוד של סדרת צעדים שיש לבצע על מנת לקבל תוצאה צפויה מראש של TC

סביבת הבדיקות: *Test Site*

- תצורת אתר הבדיקות.
- תאור סכמתי של עמדת הבדיקות.
- תאור רכיבי התוכנה באתר.
- תאור רכיבי החומרה באתר.
- זכויות, רשיונות ובעלות.
- התקנה, ניהול ובקרה של האתר ושל המערכת הנבדקת.

Test Suite

- סדרה של TC's שנועדה לכסות רכיב/פונקציה/Feature/Use Case
- הצעד האחרון בסדרה הוא לעתים התנאי לתסריט הבא

Transaction flow testing

- סדרת בדיקות לבדיקת המסלולים של תסריט או Use Case

Unit Testing

- ביצוע בדיקות ליחידת התוכנה הקטנה ביותר ברת משמעות עצמאית
- דוגמאות :סברוטינה , פונקציה , מחלקה
- בבדיקה זאת מניחים שיחידות נקראות כבר עברו בדיקה או שהן הוחלפו על ידי Stub

Usability Testing

- אמצעי שנועד למדוד את הרמה בה אנשים יכולים להשתמש ב"משהו", בהתאם ליעוד המתוכנן שלו.
- במערכת תוכנה ה"משהו" יכול להיות: מסכים,דוחות, דפי אינטרנט, מדריך למשתמש, מסכי עזרה, הודעות שגיאה, אמצעי קלט/פלט
- אם לאנשים יש קושי בהבנת הוראות, פענוח הודעות שגיאה, בצוע משימות עסקיות באמצעות המסכים , אזי המפתחים אמורים לשפר את עיצוב המערכת ולבצע מבחני Usabilityחוזרים
- במהלך ה UT המטרה היא להתבונן באנשים משתמשים במוצר בסביבה הטבעית שלהם ובצורה קרובה ככל האפשר לשימוש היום יומי, מתוך מטרה לאתר שגיאות או תחומים לשיפור

Volume Testing

- בדיקות שנועדו לבדוק את תגובת המערכת לנפחי נתונים מרביים כפי שהוגדרו במפרט הדרישות

White Box

- אסטרטגיה לביצוע בדיקות, במיוחד Unit Test
- שמות נרדפים: Structural Testing,Glass box testing
- בגישה זאת חייבת להיות גישה לתוכניות המקור