שלום!

הכנסו ל: <https://awsum.herokuapp.com/>

|  |  |
| --- | --- |
| מה מצאתם? | איך מצאתם? |
| אם מחשבים את אותו מספר- חוזר המספר ולא הסכום (יש מעקף)  אין אופציה למספרים שלילים בשדה הראשון  אין סכום>1000 בשדה הראשון  בשדה השני ניתן לכתוב כל דבר- מוגדר בstring  מקש אנטר מאפס במקום לחשב  תו ריק מתפרש כ-0 או כריק (אם שני השדות ריקים)  אפשר לבצע חיסור עם מספר שלילי  אין הגבלה על חיסור  Clear לא מנקה תוצאה, רק שדות  קצורי מקלדת, טאבים אינם לפי הסדר הרצוי  כפתורים- צבע וגודל קצת מטעים  מקש F12 נותן הערות "מענינות מאוד" | עיון בקוד המקור  הזנת קלטים- מקרי שימוש |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סוגי בדיקות | | | | |
| ידני/אוטומטי | דינמי/סטטי | פונקציונלי/לא פונקציונלי | קופסא שחורה/לבנה/אפורה | פרוגרסיה/רגרסיה/שפיות/אימות |
| ידנית- בודק אנושי מתפעל את המערכת לשם בדיקה  אוטומטית- סקריפט כלשהו מתפעל את המערכת לשם בדיקה | דינמי- המערכת הנבדקת רצה לשם הבדיקה  סטטית- אין הרצה של התוכנה הנבדקת | פונקציונלי- **מה** עושה המערכת (התאמה לדרישות)  לא פונקציונלית- **איך** שימושיות, בצועים, אבטחה, יעילות | קופסא= איך מאוחסן הקוד של המערכת הנבדקת.  קופסא שחורה-הקוד מוחבא מהבודק. ניתן לבדוק קלט ופלט.  קופסא לבנה- הקוד חשוף לבודק. ניתן לבדוק יעילות, מימוש פנימי  קופסא אפורה- לא רואים קוד- רואים "עקבות" שהקוד משאיר. לוגים, דאטה בייסים, תעבורת רשת (סניפר) | פרוגרסיה=נבדוק את כל הפיתוחים ה**חדשים** בתוכנה  רגרסיה= נסיגה. נבדוק שהפיתוחים ה**ותיקים** בתוכנה, לא נפגעו עקב עדכונים שונים, בתוכנה.  שפיות sanity- בדיקה לא מעמיקה אך מקיפה, שנועדה לוודא בשלות של המערכת לסב בדיקות מעמיק  אימות confirmation- בדיקה שבאה לוודא, שתיקון המתקבל לתקלה, אכן פתר את הבעיה. |

תהליך הבדיקות הבסיסי

קלט: התוכנה לבדיקה, אפיון (דרישות)

פלט: דיווח על מצב התוכנה הנבדקת (תקלות וכד'), דיווח על צורת הבדיקה

**Plan תכנון**

מטרת השלב הזה- להגדיר את האסטרטגיה לבדיקות. **תיחום** scope- מה ייבדק, מה לא ייבדק. **פירוק פונקציונלי**= נפרק את התוכנה הנבדקת ליחידות עבודה קטנות, אסטרטגית בדיקות- קופסא שחורה? לבנה? נכתוב אוטומציה? ועוד...לוחות זמנים, מנהלים סיכונים, קביעת **קריטריון יציאה=** נחליט באילו תנאים, נסיים את הבדיקות.

כל התכנון הנ"ל, יתועד במסך ששמו stp software test plan – באחריות מנהל הבדיקות

**Analysis and design ניתוח ועיצוב**

**ניתוח-** בשלב ניתוח- מנתחים את האפיון (הדרישות) שהגיעו לתוכנה, ומייצרים מהם "תנאי בדיקה"- הגדרה לבסיס הניתן לבדיקה. מתרגמים מסמך דרישות לכדי אלמנטים של בדיקה. לרוב, נהוג לכתבם בצורה כגון "אם..כאשר..אזי..."- תנאי בדיקה חייב להיות משפט חד משמעי, והגיוני.ותמיד נכון

|  |  |
| --- | --- |
| אפיון | תנאי בדיקה |
| "מכונת המשקאות תחזיר עודף מ10 ₪ במטבעות של 1 ₪" | **אם** נכניס 10 ש"ח  **כאשר** המכונה תזהה שיש צורך בעודף ויש מטבעות במכונה  **אזי** יוחזר עודף במטבעות של 1 ש"ח |

מטרת התהליך הזה: חשפנו דרישה "משתמעת". מטרת התהליך הזה- למצוא סתירות פנימיות, חוסרים, אי דיוקים באפיון- טרם נתחיל לבדוק

**עיצוב-** מעצבים **מקרה בדיקה**- סט הנחיות לבודק.

|  |  |
| --- | --- |
| מקרה בדיקה | |
| **צעד לבצוע** | **תוצאה רצויה** |
| **נלמד מתודולוגיות לעיצוב מקרי בדיקה-** איך בוחרים קלטים, שילובים, התקנות וכד' בצורה האופטימלית? | מתוך האפיון, ומתוך שלב הניתוח, שעשינו קודם |

התוצר של שלב ניתוח ועיצוב יתועד במסמך std software test design. שלב זה בתהליך הבדיקות הינו באחריות הבודקים.

**Implement and execution**

**יישום**

היערכות לקראת סבב בדיקות: התקנות, רשיונות, כיול כלים, ארגון סביבות עבודה, ארגון data, ארגון "מנות בדיקה"-מקבצים את מקרי הבדיקה ל"מנות" אשר ירוצו ביחד, כדי לצמצם תקורות של הקמה וארגון טרם ההרצה.

**בצוע**

הרצת הבדיקות, טיפול בתקלות, בצוע ניסויים

**Eval and report**

**הערכה**

מנטרים את התקדמות הבדיקות בכדי לדעת האם הושג "קריטריון היציאה" שהגדר בשלב התכנון

**ודיווח**

מדווחים על סיום הבדיקות- במסמך שמו str software test report- באחריות מנהל הבדיקות

**Closure**

סיום העבודה, הפקת לקחים, העברת הידע הלאה....

עבודה עם practitest