בדיקות יחידה

יתרונות:

- נכונות (באמצעות שפת בדיקות - אין כל בדיקות אוטומטיות)
- פתרון שגיאות debugging - הקוד יבודד כך שכל בדיקות ויבדקו פתרון שגיאות debugging.
- תיעוד של מציאות - אין מתאים? הגדרת קוד, קובץ טקסט וכו' בדיקות המלה לא קטנות שמתיעדות כך - רוח של בדיקות שאפשר והרבה לא הכיר אפשר לראות לא הכיר אין קוראים לסוגן בדיקה נכונה וכו'.

חסרונות: - אין בדיקה של כל המערכת ביחד

- נוקה הרבה כסף: כמה קוד קסמי ואמו שכיר מתבוקש בדיקות עם עוקה כסף.

פרקטיקה מקובלת בתעשיה: לתעד בסגנון לפני הקוד והוא אפשר ה- design של הקוד. test אנשים נכתב, כמבוים מעיצוב קוד שיבדיק את ה- test וכך הלאה..

האמירה היא שלא כיסוי טוב יותר של הקוד, תכנון מתאים - design תוך כדי, זאת לתכנן מבנים מה המערכת עשה בכל שלב.

- כשאנחנו לכתוב קוד אין כדי לבדוק אותו ולכן כדאי לכתוב בדיקות לפני.

יש פה חסרונות: • לא מסתדרים לא התאוננה הבלתי אל פרק

• נוקה הרבה כסף להתאמקבל בדיקה

• סלע כדי של הבעת מחיר שיתל פאולה מבנה הלאה.

בדיקות יחידה טובה: - כיצד לדאוג מהו

- מקודד - כל בדיקה יכולה לרוץ לבד

- ניתן הגדר אות אותה הבאמה. קלי תלות בגורמים חיצוניים כדי שהיה

נעים להגיד בקומים אם test מהר אז לא.

- יבדוק את עצמו

- בחומרת מה כמבוים test.

סקירות לבדיקות סטנדרטיות