שיטת TDD (Test-Driven Development) היא תהליך פיתוח תוכנה בו הקוד מתוכנת בהתבסס על בדיקות בכתיבה שלה. השיטה מתחילה עם כתיבת בדיקה (קוד לבדיקה) שמתארת את התנאי המבוקש לתכנית החדשה או לשיפור בתכנית קיימת. רק לאחר מכן מתבצע כתיבת הקוד של התכנית עצמה, ובסיום, מתבצעת הרצת הבדיקה שנכתבה בהתחלה. המטרה העיקרית היא לוודא שהתכנית תופעל כראוי ותקיים את התנאים שמוגדרים.

יתרונות של TDD:

1. איכות קוד גבוהה: כשבונים בדיקות במהלך פיתוח הקוד, יש יתרון בבנייתו של קוד יציב, נקי וקל לתחזוקה. הצורך לכתוב בדיקות תורם להקפצת איכות הקוד.

2. הידוק באירוע שגיאות: בדיקות מאפשרות גילוי מוקדם של בעיות ושגיאות בקוד, ולכן ניתן לתקן אותם בטרם הם מתרחשים בסביבת הפרודקשן.

3. יכולת לשדרג קוד קיים: כאשר כל חלק מהקוד מקושר לבדיקה, השדרוגים והשיפורים נעשים בצורה בטוחה ולא מעמידים פעם נוספת את הקוד בסכנת פעילות אחרי שדרוג.

4. תיעוד מובנה: כשהבדיקות מספקות דוגמאות מתועדות לשימוש בקוד, זה מסייע בכתיבת התיעוד ובהבנה יותר מהירה על ידי מפתחים אחרים.

5. חסכון זמן: למרות שבתחילה זה עשוי להיראות כמו יצירת עומס נוסף בפיתוח, הבדיקות מסייעות לזהות בעיות קוד מוקדם, וזה עוזר לחסוך זמן ומאמץ בהמשך.

חסרונות של TDD:

1. הדרך הנכונה לעבוד: עשוי להיות מאתגר להחליט על הסדר התפתחות הקוד המדויק שאליו יתבצע הבדיקות בכדי להתקדם בפיתוח.

2. הקוד של הבדיקות עצמן: ייתכן ולפעמים הקוד של הבדיקות עצמן יכול להפוך לקוד מסורבל וקשה לתחזוקה.

3. זמן ומאמץ: הקפדה על כתיבת בדיקות עשוי לדרוש זמן ומאמץ נוסף בתהליך הפיתוח.

4. אינטגרציה עם סביבות חיצוניות: בפעמים, ייתכן ותהליך הפיתוח מתבצע בסביבת פיתוח אחרת מאשר סביבת ההפעלה הסופית של התוכנה, ולכן ייתכן והבדיקות לא יתבצעו באופן מדויק בסביבת הפרודקשן.