נכתב ע"י: אביה צ'ריקר :)

הנדסת תוכנה שיעור 6

בקרת תצורה git/github:

שאלת השיעור: בניה – סבבי פיתוח

הנדרש: עבודה על אותו קוד של כמה מפתחים, שמירת שינויים, בקרה אחר הגירסאות השונות.

הכלי: כלים לבקרת גירסאות כגון github

לפרויקט תוכנה תוצרים שונים:

- מסמכי דרישות ותיכון, קוד, executables ,מדריכי שימוש, בדיקות וכו'
- פרויקט תוכנה משתמש בכלים שונים: מהדרים, עורכים, צד ג', שת"פ וכו'

למה משתמשים בבקרת תצורה?

- איסוף כל הגרסאות ומעקב אחרי השינויים
  - חזרה לגרסה מסוימת (במקרה שיש תקלות או באגים)
  - השוואה
  - מאפשר מחיקת קוד ללא חשש – הרי תמיד אפשר לחזור אחורה
- ניהול מספר גרסאות במקביל (יכול להיות מספר מערכות הפעלה שצריך לתמוך בהם, יכול להיות שיהיו כמה לקוחות, מאפשר לכל אחד לעבוד על משהו אחר במקביל וכו')
- גיבוי והצלה
- שיתוף מספר מפתחים (מרוחקים) בו זמנית (טיפול בסתירות)
- מאגר מעודכן של תוצרי הפרויקט

בקרת שינוי –

1. זיהוי ותיעוד – למשל מי משנה, הסיבה, זמן וכו'
2. ניתוח והערכה של שינויים - אנחנו רוצים כלים שיאפשרו לנו לחזור אחורה לעשות השוואה של דברים
3. אישור/דחיה של שינויים – האפשרות להחליט אם שינוי נכנס או לא
4. אימות, מימוש ושחרור

בקרת גירסאות

- מאגר
- הכנסה והוצאה
- ענפים ומיזוגים – אפשר שלאותו מאגר יהיו כמה גרסאות מקבילות
- תיוג

נכתב ע"י: אביה צ'ריקר :)

כלים: היסטוריה (השוואה)

המשותף לכל הכלים שהיו ב15 השנה הראשונות היו הריכוז – היה שרת משותף שכולם עובדים דרכו והכל נשמר בו

בשנים האחרונות – לכל אחד יש את ההיסטוריה על המחשב שלו

החסרונות בגיט:

- שמישות – המקור גיט זה בסה"כ אוסף של כמה כלים, וקשה לזכור את כל הפלאגים
- עד לפני כמה שנים היו בעיות בגיט בנושא קבצים גדולים – כבר פתחו
- בנושא של נעילות – זה נושא חלש בגיט ולכן לארגונים גדולים קשה להשתמש בה

משל גיט:

נסביר מלמטה למעלה איך גיט עובד

במשל הזה כל מה שאנחנו צריכים זה מחשב ועליו:

1. עורך טקסט
2. פייל סיסטם – לנהל את הדברים

המטרה היא לכתוב תוכנית גדולה כאשר אנו רוצים להשתמש בכלים לבקרת גרסאות

עם השנים אנחנו צוברים תמונות כדי לראות איך אנחנו נראינו וכו', אותו דבר בכתיבת תוכנה – נרצה מידי פעם לעשות עותק של דברים שכתבנו ליתר ביטחון, הגרסאות השונות מתמזגות עפ"י תאריכים,

זה מה שאנחנו עושים, הלקוחות מרוצים, אנחנו ממשיכים לפתח, והלקוחות פתאום לא מרוצים כי משהו לא עובד – מה עושים? חוזרים אחורה לגרסאות קודמות שעבדו ומתקנים עפ"י השוואות וכו'

את השינוי נשמור בגרסה נוספת הצמודה לגרסה שבה הייתה הבעיה.

חוץ מהענפים שגדלים כמו עץ כמו קומיטים, יש גרסאות שאנו רוצים לשמור אותם בגרסאות חשובות.

בדרישות הקורס שלנו – כל קוד שמשחררים ללקוח לשימוש לתייג כגרסה חשובה תחת כותרת כדוגמת : "שחרור קוד ללקוח של איטרציה X".

הקוד נשמר בפונקצית האש.

הבעיות בזה –

1. התנגשויות – פחות נפוץ.

2. פחות קריא לבני אדם.

נניח ש-2 אנשים עובדים על ענפים שונים , איזה מהענפים נשמור? הפתרון – מיזוג – נמזג את כל השינויים ונשמור כגרסה אחת.

git pull:

עושה git fetch – לוקח את כל ההיסטוריה וממזג אותה.

אח"כ עושה git merge - ממזג את הקוד שלנו איתה.