שיתוף וניהול קבצים, מבנה ספריות לפרויקט

1. אנו מתחילים את הפרויקט, והדרך הטובה ביותר להעביר ולנהל קבצים היא לא באמצעות המייל, אלא דרך תוכנה לשיתוף וניהול קבצים, דוגמת SVN (מייל נפרד יישלח בנוגע להתקנה התוכנה).

כלי ה- SVN הוא כלי לבקרה וניהול תצורה של מערכת קבצים. הוא שימושי בעיקר לניהול פרויקטים הכוללים פיתוח קוד (תוכנה, VHDL...), כאשר הוא מאפשר שמירה של מערכת קבצים על כל גרסאותיה. בנוסף, הוא יכול לשמש לשמירה וניהול תצורה של כל סוגי הקבצים (בינארי / טקסטואלי).

הכלי מאפשר לבצע מעקב אחר שינויים ולנייד קבצים (ולא לשלוח אותם דרך המייל או לשתף אותם בכונן ייעודי).

עיקרי יכולות הכלי הן:

* מהווה מערכת גיבוי לקבצים, ומונע איבוד מידע במקרה שהקבצים במחשב האישי נמחקו מסיבה כלשהי (לדוגמא - תקלה בכונן המקומי).
* מאפשר מעקב אחורה לגרסאות קודמות של הקובץ. –דוגמא לחשיבות יכולת זו: כאשר מוסיפים יכולת למודול, מסתבר שיכולות קודמות נפגמו. שיטוט בשינויים מגרסאות קודמות יכול לעזור להבין מה גרם לבעיה.
* תומך בניהול גרסאות ושחרור רשמי ע"י הוספת תווית (label).
* מאפשר מעקב לא רק לקובץ בודד, אלא לפרויקט כולו (מה השתנה, מה חסר ב- SVN, מה חסר בכונן המקומי).
* מאפשר שיתוף קבצים בין מתכננים שונים בצורה נוחה. כשמוסיפים קובץ למאגר המידע של הכלי הוא שקוף לכולם (ניתן גם לקבוע הרשאות – מי רשאי לערוך, לצפות, לקחת...).

1. רצוי לקבוע מעתה מבנה ספריות לפרויקט. כדי למנוע מצב שהכול זרוק בספריות

בצורה די שטוחה וללא סדר הגיוני, אני ממליץ ליצור שכבה ראשית של ספריות ע"פ סוגי קבצים, ושמה

להמשיך עם ספריות ע"פ פונקציונאליות וכך הלאה. כלומר להכניס עומק למבנה הספריות.

הספריות הראשיות שאני ממליץ שיהיו הן:

VHDL

DOC

Presentation

MATLAB

EXCEL

Data\_Sheet

בתוך VHDL יהיו שתי ספריות ראשיות: DESIGN ו- SIMULATION. בתוך כל אחת

מהן תהיה חלוקה ע"פ הפונקציונאליות.

בתוך DOC יכול להיות הרבה ספריות. לדוגמא: Meeting\_Summary, Project\_Doc,

Moshe\_Commnets......

בתוך Presentation תהיה חלוקה ל- Characterization, MidTerm ו- Final.

וכך הלאה.

מעבר לסדר, הדבר מקל על הניווט והשיטוט אם מאמצים זאת בכל מיני פרויקטים, וגם כאשר שומרים קבצים במחשבים שונים.

הוספתי דוגמא חלקית בעמוד הבא.

דוגמא מלאה ניתן למצוא ב- <http://runlen.googlecode.com/svn/>

משה