**ביקורת מערכות מידע - שיעור 1**

**יואב פיאטקובסקי**

**טלפון : 054-4708200 – חירום**

**טלפון 2 : 054-4878085 – יפית**

**מייל:** [**pjoav@zahav.net.il**](mailto:pjoav@zahav.net.il)

ספר הקורס :

1. 4 ספרים שחורים של פרופ' הרצל פטל
2. תקני ביקורת של לשכת רו"ח – יש לרכוש 300 ₪ מהדורה שתצא במרץ-אפריל 2015

**פרק 1 - מבוא**

אנו לומדים כעת סט של הרצאות שעוסק בתחום של ביקורת מ"מ

**1.1 רקע היסטורי כללי**

תקן ביקורת 66 היה תקן הביקורת הראשוני והתשתיתי שעסק בתחום של מ"מ כאשר תפישת העולם היתה שיש הבחנה בין ביקורת מ"מ לביקורת קלאסית. התפישה וההתייחסות אל ביקורת מ"מ היתה בנפרד לביקורת הרגילה . התפישה הזאת היתה נכונה לשנות ה-90 של המאה הקודמת. 20 שנה זהו זמן רב בעיקר במונחי מחשבים . כשנכתב התקן עדיין לא היו קיימים אמצעים טכנולוגיים מתקדמים שקיימים כיום . בתקן היו קיימים נספחים מיוחדים שהסבירו מושגים שהיום הם נראים לנו פשוטים ובסיסיים שאז לא היו ברורים לרב האנשים. תפישת העולם בנושא השתנתה לגמרי. עולם הביקורת השתנתה שינוי מבורך וכיום ביקורת מ"מ מהווה חלק מהביקורת הכללית. ת"ב 66 בוטל ולא החליף אותו אף ת"ב ספציפי שעוסק במ"מ אלא שביקורת מ"מ מהווה חלק מהביקורת הרגילה. לדוגמא: בת"ב 93 היה פרק שמתייחס למ"מ למשל מודל הקוסו . ואחת מהשכבות במודל הקוסו דיברה ספציפית על מ"מ , בת"ב 82 נטען שחלק מראיות הביקורת ניתן לאסוף באמצעים ממוחשבים, בת"ב 85 כתוב שניתן לבצע דגימה באמצעים ממוחשבים וכו'....ביקורת מ"מ כיום משולבת בילט-אין בביקורת הכללית .

ישנם שני ת"ב ספציפיים שנכתבו רק לביקורת מ"מ שהם:

1. **ת"ב 94** – תקשורת עם לשכות שירות
2. **ת"ב 95** – סחר אלקטרוני ובעצם איך רו"ח מיישם תקני ביקורת בסביבת מסחר אינטרנטי.

ישנם עוד כל מיני דברים שהם אינם תקני ביקורת שמשפיעים על מ"מ ומשפיעים עלינו כמבקרים כי הם מעצבים את מבנה מ"מ.

למשל ישנו פרסום מאד חשוב שנקרא **הוראות ניהול ספרים של מ"ה .** הוראות ניהול ספרים מעניינות אותנו כמבקרים כי הן מחייבות כל חברה לגבי איזה ספרי חשבונות עליה לנהל, אילו נתונים יהיו בספרי החשבונות, אילו מסמכים הנה"ח צריכה להפיק , מה יהיה תוכנם , איפה ישמרו וכמה זמן, אילו גיבויים יש לבצע וכו'...

או למשל **חוק הגנת הפרטיות** שקובע ומעצב כיצד תיראה מ"מ בארגון – אם אני אוסיף אינפורמציה עלי לאבטח אותה, אסור שפרטים אישיים של בן אדם יועברו למקומות שהוא לא נתן את הסכמתו לכך שהם יקבלו מידע עליו. זוהי בגדול המסגרת שלנו .

**1.2 מושגי יסוד במערכות מידע**

**"היררכיית הנתונים" –** איך נתונים נאגרים ונאספים- הילאבטח אותה, אסור שפרטים אישיים של בן אדם יועברו למקומות שהוא לא נתן את הסכמתו לכך שהם יקבלו מידע עליו. חידה הבסיסית של נתונים היא תו (8 ביט), כל אות מהווה תו , אוסף של תווים שיש ביניהם קשר זה שדה , אוסף של שדות שיש ביניהם קשר הוא רשומה , קובץ הוא אוסף של רשומות בעלות מכנה משותף, מעבר לקובץ- מסד נתונים הוא אוסף של קבצים המאוגדים ביניהם ביחד (DATA-BASE).

**"שיטות עיבוד"** – קיימות שתי שיטות עיבוד :

1. **BATCH-** כשמדברים על באטצ' אנו בעצם מעדכנים את מערכת הנתונים במנות בנגלות. אחת לכמה זמן מעדכנים למערכת סט או קבוצת נתונים כמו חשבשבת- עדכון קבצים ללא הפסקה וכאשר סוגרים את הקובץ לא ניתן לעדכן אותו יותר
2. **REALTIME-** המערכת מבצעת עדכון עבור כל נתון שמוזן אליה למשל: נכנסתי לאינטרנט וראיתי אולם תיאטרון ואני רוצה להזמין כרטיסים להצגה ברגע שאני מעדכנת מקומות באולם ושילמתי עבור אותם מקומות לא ניתן לבטל והמערכת מתעדכנת ואותם מקומות מוצגים כתפוסים באופן מיידי.

ההבחנה ביניהם חשובה והכרחית לנו בכל מיני מונחים של מ"מ למשל: בתחום של בקרות קלט צריך מ"מ מאובטחות יותר במ"מ בשיטת זמן אמת כי שם אי אפשר לתקן .

קיימות גם מ"מ משולבות כמו:

**עיבוד צל** – זוהי הכלאה בין באטצ' לבין רילטיים – המערכת התפעולית היא מערכת באטצ' אבל המערכת שמוצגת למשתמש היא מערכת אונליין . סוג כזה של עיבוד משמש בעיקר בבנקים **למשל:** הלכתי לכספומט משכתי 200 ₪ כעת כשאעשה בירור יתרה תופיע עדיין היתרה לפני המשיכה הבנקים מבצעים עיבוד לילי ובמסגרת העיבוד הלילי מעדכנים את כל הנתונים שהתעדכנו במהלך היום. למשתמש הרגיל זוהי מ"מ בשיטת הרילטיים כיוון שברגע שאקיש סכום שאינו נכון בכספומט עדיין יצא לי הסכום השגוי ז"א בזמן אמת זוהי מערכת אל חזור למרות שהמערכת תדעדכן רק למחרת.

**"ארכיטקטורה של מ"מ" –** אנו יכולים לעבוד במס' שיטות עבודה :

1. **מבנה ריכוזי –** יש מחשב מרכזי ומסופים שמחוברים אליו ניתן לראות זאת בבנק , במ"ה , ביטוח לאומי וכדו' ...
2. **צורת עבודה ביזורית –** עבודה עם שרתים מקומיים ובעצם כל קבוצת מחשבים פועלת באופן עצמאי ומבצעת את העיבודים ללא תלות במערכות אחרות
3. **השיטה המעורבת (ריכוזי חלקי)-** חלק מהעיבוד במחשב מרכזי וחלק מהעיבוד בשרתים מקומיים כמו אחסון ועיבוד של נתונים. שיטה זו מיועדת להפחית עומס מהמחשב המרכזי.

**1.3 סוגי מערכות מידע**

1. **מערכות מידע ניהוליות –** מערכות מידע המשמשות לצרכים ניהוליים בארגון למשל: מערכות לניהול מלאי , משאבי אנוש הנה"ח... תפקידן הוא לספק מידע למנהלים בתחומים השונים
2. **מערכות תומכות החלטה** - מערכות מידע המסייעות למנהלים ובכירים בארגון לקבל החלטות – מערכות שמבצעות תחשיבים וניתוחי רגישות המערכת הידועה ביותר היא אקסל והיא המערכת הבסיסית ביותר לניתוחי רגישות.
3. **מערכות מומחה** – מערכות שמקבלות החלטות מובנות, מערכת של IF שברגע שעובדים איתה המערכת יודעת להגיע לתוצאה הנכונה לפי הפרמטרים שאני מציב בה.

באופן עקרוני ככל שבעל התפקיד בארגון הוא יותר בכיר הוא צריך מידע פחות ממוקד, יותר כללי וברזולוציה פחות חזקה ובתדירות יותר נמוכה . ככל שאדם בדרגה ניהולית יותר נמוכה הוא זקוק למידע בתדירות יותר גבוהה ובצורה ממוקדת יותר. **למשל:** את רמי לוי עצמו לא מעניין כמה קרטוני ביצים יש בסניף במגדל העמק וכדו' אבל הרבה יותר מעניין אותו מה מחזור הסניף שלו בקריית אתא, ומה פלח השוק שלו לעומת מתחרים... ועוד כל מיני נתונים מהסגנון הזה עליהם הוא מסתכל על אחת לתקופה. לעומתו מנהל הסניף של רמי לוי במגדל העמק מאד מעניין כמה קרטוני ביצים הגיעו לסניף ואילו כמות המכירות של רמי לוי באיזור הצפון ממש לא מעניין אותו . ז"א ככל שיורדים בדרג התפקידים המידע צריך להיות זמין יותר וממוקד יותר

**מסקנה :** מ"מ צריכה להתאים למשתמשים השונים ויכול להיות שבארגון מסויים יהיו מספר מ"מ שכל אחת תתאים למשתמשים שונים.

**1.4 מיהם המשתמשים במערכות המידע?**

יש לנו 2 קבוצות של משתמשים :

1. **משתמשים חיצוניים-** המשתמש הכי כבר ודרמטי במ"מ של הארגון הוא רשויות המס וביטוח לאומי . כל חודש עורכים דו"ח מע"מ , מעבירים ניכויים וכו' ... מאיפה שואבת רשות המיסים את האינפורמציה – רק מתוך מ"מ של הארגון , בנוסף גם בנקים , ספקים, קוראי הדוכ"ס ו**רו"ח**.

* **רו"ח הוא חיצוני לחברה אנחנו לא חלק מהמערך האירגוני בחברה כשאנו באים לערוך את הביקורת של החברה אנו שואבים את המידע מתוך מ"מ של הארגון אנו חיצוניים לחברה.**

1. **משתמשים פנימיים –** בעלי התפקידים לפי צרכיהם השונים בארגון כמו : מנהל משאבי אנוש שצריך להסתכל על נתונים של עובדים או מנהל קשרי לקוחות שצריך לעבוד על חוזים שחתמה החברה עם לקוחות**.**

**1.5 מבנה יחידת ה-IT בארגון:**

(עמוד 9-10 בחוברת של אייל אסייג) מדובר פה על מבנה של ארגון גדול כמו למשל: בנק, חברת חשמל, תעשייה אווירית ככל שהארגון קטן המבנה הופך לשטוח יותר ובכל ארגון מבנה ה-IT שונה. חלק מהדברים עוסקים בפיתוח ותכנות אבל אם אני למשל בעל מפעל באיזור התעשייה בלוד מחזיק 30 עובדים מחזור שנתי 40 מיליון ₪ בשנה אני לא זקוק למתכנתים סביר להניח כי אשתמש בתוכנות מדף תוכנות מובנות לא תוכנות שיפתחו בשבילי לעומת זאת אורנג' שהיא חברה ענקית יש צורך במתכנתים לצורך בניית תוכנות שמתאימות לצרכיו של הארגון. בראש מחלקת IT עומד סמנכ"ל IT שאחראי על התקציב של היחידה על חלוקת התקציב, תיעדוף, מה עושים מתי וכדו'. למשל: האם נחליף מחשבים בהנה"ח , האם נשנה את מ"מ של המחסן וכו'

מחלקת ה-IT בארגון מתחלקת לשתי קבוצות:

1. **מחלקת פיתוח –** עובדת בסביבה שונה לחלוטין מסביבת המשתמש.

(כאשר אנו מדברים על סביבת המשתמש אנו מדברים על **סביבת הייצור** המקום בו עובדים האנשים המשתמשים בתוכנה. למתכנתים יש סביבה שונה לחלוטין שנקראת **סביבת פיתוח**, הם מנותקים לגמרי מהמשתמשים, האייקונים שהם משתמשים בהם שונים בתכלית מהאייקונים בהם אנו משתמשים. בין סביבת הייצור לסביבת הפתוח יש סביבת ביניים שנקראת **סביבת בדיקה – ניסוי** שהיא בעצם השלב שבו בודקים בתנאים מבודדים ומבוקרים את התוצר של סביבת הייצור לצורך בדיקת איכות התוכנה לבדוק שהתוכנה עובדת ואין בה באגים וכאשר הגענו למסקנה שהכל עובד בצורה תקינה ניתן להעביר את התוכנה למשתמש הסופי.) במחלקה זו יש 2 פונקציות מרכזיות:

1. **מנתחי המערכות –** בתור משתמשים אנו באים למנתח המערכות ומסבירים לו איזו מערכת מידע נדרשת לנו הוא בעצם מגשר בינינו לבין המתכנתים והוא בעצם ממיר את מה שאמרנו לו בע"פ ובשפה פשוטה לשפת תכנות שמתאימה למתכנתים שהם בעצם כותבים את התוכנה והם בודקים את יכולת הביצוע של אותה מ"מ וחוזרים אלינו עם ניתוח המתכנתים ומסבירים מה ניתן או לא ניתן לעשות.
2. **מתכנתים -** מקבלים ממנתח המערכות את התרשימים ועוברים עליהם ובודקים איך ניתן ואם ניתן לבצע את דרישת הלקוח **.**
3. **מחלקת תשתיות-** מחלקה זו מתחלקת למספר בעלי תפקידים:
4. **מנהל הרשת –** בכל ארגון שמכבד אץ עצמו ישנו מנהל רשת שדואג להרשאות ולתפעול הרשת, הגדרת משתמשים , אבטחת מידע ברשת
5. **DBA – DATA BASE ADMINITRATOR-** אחראי לנושא מהירות העברת הנתונים, טיוב הנתונים, מיון הנתונים, עדכון הנתונים במערכת, עדכון מערכת המידע, הוא עושה את זה בצורה שוטפת ובעצם מתחזק את המערכת ומונע כפילויות.
6. **מפעיל מחשב –** אחראי לגיבוי השרת של כלל החברה , מעדכן גרסאות וכו'...
7. **תמיכה טכנית –** אנשים שפותרים תקלות במערכות ההפעלה כי לא בשביל כל תקלה מזמינים מומחים חיצוניים לצורך חיסכון בזמן ובעלויות.
8. **HELP DESK – חדר מיון –** נותני השירות הראשוני ללקוח, למשתמש במערכת.
9. **קצין אבטחת המידע –** מעין המבקר הפנימי של מחלקת ה-IT אותו פרנציפ עובד גם בהקשר של קצין אבטחת מידע אותו אדם הוא האחראי המפקח הבודק של האבטחה הלוגית והפיזית של מחלקת ה-IT מזמין מומחים לביצוע בדיקות חדירה למערכת , מבצע ניתוחי רגישות עצביים במערכת לאיתור תנועות חריגות . בכל מערכת יש **יומן Log** - שעובד כמו מצלמת אבטחה ומצלם כל תנועה שנערכת על המחשב איזה תוכנות הפעלת וכדו'... קצין אבטחת מידע אחראי על היומן הזה והוא בעצם עוקב אחר תנועות חריגות ובודק אותן. בנוסף קצין אבטחת מידע עורך גם בדיקת תקינות של CRUD

**CRUD-** זה ראשי התיבות למערך ההרשאות בכל מ"מ:

**C**- CREATE הזכות ליצירת רשומה

**R**-READ ONLY – הזכות לקרוא רשומה

**U**-UPDATE- הזכות לעדכון רשומה

**D**-DELETE- הזכות למחוק רשומה

**פרק 2 – מחזור חיים של מערכת מידע**

**2.1 שלב הייזום**

למה בכלל שנרצה לעשות ייזום ? למה זה טוב ? יש לי מערכת שאני עובד איתה למה ייזום ? – אם המערכת נתקעת , עצלה , המסכים לא זזים ואני מרגיש שהמערכת ישנה לי אז צריך להחליף את המערכת. לפעמים אני יכול להחליף מערכת כתוצאה משינוי שהוא כלל ארגוני **למשל:** יש לי תוכנת שכר מעולה אבל התקבלה החלטה של מנהל ה-IT שאין יותר מערכות נפרדות אלא כל הארגון עובר למערכת TRP כללית ז"א יכול להיות שהסיבה לשיקול להחליף מ"מ היא לא כי המערכת הספציפית לא טובה אלא מתוך דרישה חיצונית או חוקית לעבור למערכת אחרת. בד"כ כאשר אנו נמצאים בשלב הייזום החברה תוציא מסמך שנקרא **מסמך ייזום- RFI- REQUEST FOR INFORMATION**- אין לי בעיה מוגדרת אני יודע שהמערכת הקיימת לא טובה לי אין לי תכניות מוגדרות ואני מציע הצעה לקבל רעיונות למ"מ חדשה . כל הדברים הללו שדיברנו עליהם עכשיו זה דברים שהחברה עושה .

אנו בתור רו"ח נדרשים לבצע הערכת סיכונים כללית רב שנתית ואנו עושים תיעדוף אילו מ"מ אנו צריכים לבדוק באיזה סדר ובאיזה עדיפות לבדוק כל מערכת החלטות בארגון לגבי החלפת מערכת משפיעה על תיעדוף הבדיקה של מ"מ. למשל אם החברה רוצה להחליף את המערכת צריך לבדוק למה החברה רוצה להחליף מערכת האם יש צורך וסיבה לבדוק מערכת שממילא אוטוטו מוחלפת וכו' .

**2.2 חקר היישימות**

בשלב זה אנו בודקים לא רק ברמת התיאוריה אלא ברמת הפרקטיקה האם ניתן לבצע את המערכת החדשה כשמדברים על חקר היישימות צריכים לבדוק כמה דברים שצריכים להתקיים בקשר של גם וצריך לעמוד בכל הפרמטרים פגיעה באחד מהם תפיל את המערכת:

1. ישימות כלכלית- האם אני ערוך מבחינה תקציבית להיכנס להרפתקה הזאת שנקראת מערכת חדשה. אז לפעמים יש לי רעיון אדיר ואני רוצה נורא לעשות משהו אבל חוסר ישימות כלכלית מונעת את ביצוע הרעיון.
2. ישימות טכנולוגית – לא ניתן לבצע את הרעיון מבחינה טכנולוגית למשל : לפני 20 שנה עשו לסטודנטים תרגיל כולם הבריזו מהשיעור והשאירו למרצה מכשירי הקלטה שיעור אח"כ המרצה הביא טייפ והשאיר על השולחן דולק. היום יש אמצעים טכנולוגיים מתקדמים יותר אבל ייתכן שחלק מהדברים לא ישימים בנק' זמן מסויימת.
3. ישימות ארגונית – האם הארגון ערוך לקבלת המערכת החדשה לפעמים יש את הטכנולוגיה ואת האמצעים הכלכליים והארגון לא ערוך לקבלת המערכת והיא תיכשל
4. עמידה בלו"ז- יש משמעות גם ללוח הזמנים ולעיתוי בו המערכת תוכל להיכנס לפעולה. אם אני לא עושה את זה בטווח זמן הגיוני ורלוונטי זה לא שווה כלום.

מחקרים מוכיחים ש-85% מהפרויקטים שעברו בהצלחה את חקר הישימות הצליחו וזה מאד חשוב כי ברגע שעברתי את חקר הישימות הרעיון הופך מרעיון למציאות ובשלב זה אני כרו"ח מגיע כי כשרו"ח מגיע ורואה שהמערכת עברה את חקר הישימות זה משפיע באופן וודאי על התיעדוף ועל סקר הסיכונים כלומר, במקרה כזה ברור לנו שהמערכת הקיימת תוחלף במערכת חדשה בסיכויים גבוהים מאד ז"א שאין צורך לעשות בדיקות בקרה על המערכת הקיימת כיוון שבכל מקרה הולכים להחליף אותה אבל מן העבר השני נמשיך לבצע בדיקות מבססות על המערכת הקיימת כיוון שהמידע שמגיע אלינו כרגע הוא מתוך המערכת הקיימת ולכן את הבדיקות המבססות אני עדיין אערוך במערכת הישנה

**2.3 שלב הגדרת הדרישות**

זהו השלב שבו החברה מוציאה מסמך שנקרא RFP (REQUEST FOR PROPOSAL) כאן אני יודע בדיוק מה אני רוצה. בשלב זה שלב הגדרת הדרישות מתערב הרו"ח ואותו מעניין רק היבטי הבקרות לא המבנה לא העיצוב רק הבקרות . האם החישובים נכונים האם מתבצע העיבוד בהתאם להוראות, האם התוצאות נכונות , פה המקום שלי כרו"ח לבוא ולייעץ לחברה מה לעשות לגבי הבקרות הרלוונטיות. **מודל הבטון-** אם אני מגיע לחברה שיש לה מ"מ קיימת ולא טוב לי הבקרות החברה כמו בטון יבש לא תזוז לא תיתן לי לשנות במערכת הבנויה כבר . לעומת זאת אם עכשיו הולכים לפתח תוכנה חדשה החברה הרבבה יותר גמישה ופה המקום לדרוש לתקן ולהוסיף בקרות מתאימות. הקו שמפריד בין ייעוץ לבין קבלת החלטות הוא דק מאד והמטרה של רו"ח לכוון את החברה לעבר המטרה .

**לדוגמא :** אנו רוצים לרכוש או לפתח תוכנה חדשה.

**סט תכונות משקלות:**

מחיר – 30%

איכות – 20%

מוניטין ספק – 10%

התאמה לטכנולוגיה – 20%

תפוצה – 10%

גמישות לשדרוגים בעתיד – 10%

כעת אני לוקח כל הצעה ומדרג את התכונות שלה מ-1 עד 5 ומשקלל את כל התכונות כפול האחוז שנתתי להן ומי שמקבל את הציון הגבוה ביותר אותו אקח.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | ספק A | | ספק B | | ספק C | | ספק D | |
| תכונה | משקל | ציון | סה"כ | ציון | סה"כ | ציון | סה"כ | ציון | סה"כ |
| מחיר | 30 | 4 | 120 | 2 | 60 | 5 | 150 | 3 | 90 |
| איכות | 20 | 3 | 60 | 2 | 40 | 4 | 80 | 3 | 60 |
| מוניטין ספק | 10 | 5 | 50 | 3 | 30 | 5 | 50 | 3 | 30 |
| התאמה לטכנולוגיה | 20 | 2 | 40 | 5 | 100 | 1 | 20 | 5 | 100 |
| תפוצה | 10 | 3 | 30 | 1 | 10 | 5 | 50 | 5 | 50 |
| גמישות לשינויים | 10 | 2 | 20 | 4 | 40 | 4 | 40 | 1 | 10 |
|  |  |  | **320** |  | **280** |  | **390** |  | **340** |

כעת יש לבחור באיזה תוכנה ובאיזה מצב להשתמש ויש לנו 3 אופציות:

1. **תוכנת מדף – מה היתרון בתוכנת מדף ?** – קלה להטמעה , זולה, נגישה , הסיכוי שתיהיה לנו בעיה בתוכנה הוא מאד נמוך כיוון שיש משתמשים רבים ויש סיכוי גבוה מאד שיעלו על הטעות**. מה החסרונות בתוכנת מדף?** – היא לא מותאמת אישית אני מתפשר, בשונה מתוכנה שתפורה ספציפית לחברה שמתאימה במדויק לצרכים שלי
2. **פיתוח ע"י בית תוכנה חיצוני –** תפורה ספציפית לחברה בנויה לפי הדרישות שלי אבל יש בה גם חסרונות והם – היא מאד יקרה , כמות הבאגים גבוהה, בעיית זמן כיוון שהתכנון והפיתוח אורכים המון זמן .
3. **פיתוח עצמי-** כמו הקודם - תפורה ספציפית לחברה בנויה לפי הדרישות שלי אבל יש בה גם חסרונות והם – היא מאד יקרה , כמות הבאגים גבוהה, בעיית זמן כיוון שהתכנון והפיתוח אורכים המון זמן .

**ברגע שהחלטנו באיזה תוכנה נבצע את המערכת זה משפיע ישירות על אורך החיים של המערכת :**

אם בחרתי בתוכנת מדף השלבים הבאים במחזור החיים של המערכת **לא רלוונטיים**

אם בחרתי בשתי החלופות האחרונות שהן פיתוח ע"י בית תוכנה או פיתוח עצמי אנו נתקדם בשלבים הבאים במחזור החיים של המערכת .

**2.4 שלב העיצוב**

שלב בו מקבלים החלטות לגבי איך תיראה המערכת , פונטים , מסכים, צבעים, טבלאות, מעבר בין מסכים, תפריטים ועוד כהנה וכהנה.

**2.5 שלב הניתוח**

בשלב זה מנתחים ובונים את המערכת ע"י המתכנתים, תרשימי DFD , תרשימי זרימה אחרים, עיצובים נוספים וכדו' .

שני השלבים האחרונים (עיצוב וניתוח) לא נוגעים לרו"ח ולכן לא נפרט עליהם.

**2.6 שלב התכנות**

תהליך התכנות מתבצע בצורה הבאה:

יש לנו כמה מתכנתים כאשר התהליך הוא שכל מתכנת מבצע את המשימה שלו את קטע הקוד הספציפי שלו. אח"כ מגיע האינטגרטור מלשון אינטגרלי שמאחד את כל קטעי הקוד לתוכנה אחת שלמה כל מתכנת מוסיף לדפי התוכנה שהוא כותב דפי הסבר. אח"כ מגיע הקטע המתיש של הדי-באגים שתפקידם הוא למצוא את כל הבאגים בתוכנה ולתקן אותם. אותי כרו"ח לא מעניין כיצד רושמים את התוכנה אלא מעניינת אותי הפרוצדורה כדי לוודא שהתוכנה נרשמה כהלכה והיא מהימנה וניתן להסתמך עליה , לבדוק שיש בדיקה של האינטגרטור וחתימה , האם יש בדיקות של די-באגים ודו"ח תקלות וכו' .

**2.7 בדיקות קבלה**

**טכניקת החול בעין –** אם אני אזרוק לך חול בעין היא תיצרוב ותדמע . אם אני אזרוק לך חול בעין והיא לא תצרוב ולא תדמע משהו לא תקין אותו הדבר כאן יש סט של פעולות שאני כרו"ח יודע מה צריכה להיות התגובה של המערכת אליהן ואם המערכת לא מגיבה באותה צורה משהו לא תקין במערכת למשל: אם אני מסתכל על תוכנת שכר ואני רוצה להזין שכר לעובד שנפטר או יצא לגמלאות והמערכת מאפשרת להוציא לו תלוש יש כאן תקלה . אם אני בתוכנת הנה"ח ואני רוצה להוסיף מס' ספק למערכת ואני משתמש במספר של ספק קיים המערכת צריכה להוציא לי EROR אני כרו"ח מתקיל את המערכת בכוונה כדי לבדוק את תגובותיה. אם יש תקלות יש לנהל יומן תקלות ולעקוב ולבדוק שהתקלות לא חוזרות על עצמן.

**כל השלבים שעד עכשיו דיברנו אודותיהם הם שלבים הקשורים לפיתוח ע"י בית תוכנה חיצוני או פיתוח עצמי בתוכנת מדף טופלו כל הבעיות הללו לפני שהתוכנה הגיעה אלי.**

**2.8 שלב ההסבה**