

הבטחת איכות תוכנה – מעקרונות ליישום

דניאל גלין



עריכה ועיצוב פנים ועטיפה: **שרה עמיהוד**

שמות מסחריים

שמות המוצרים והשירותים המוזכרים בספר הינם שמות מסחריים רשומים של החברות שלהם. הוצאת הוד-עמי והמחבר עשו כמיטב יכולתם למסור מידע אודות השמות המסחריים המוזכרים בספר זה ולציין את שמות החברות, המוצרים והשירותים. שמות מסחריים רשומים (registered trademarks) המוזכרים בספר צוינו בהתאמה.

הודעה

ספר זה מיועד לתת מידע אודות מוצרים שונים. נעשו מאמצים רבים לגרום לכך שהספר יהיה שלם ואמין ככל שניתן, אך אין משתמעת מכך כל אחריות שהיא.

המידע ניתן "כמות שהוא" ("as is"). הוצאת הוד-עמי והמחבר אינם אחראים כלפי יחיד או ארגון עבור כל אובדן או נזק אשר ייגרם, אם ייגרם, מהמידע שבספר זה, או מהמדיה (אם תצורף), או מאתר האינטרנט המלווה את הספר (אם יהיה).

אין לעשות שימוש מסחרי ו/או להעתיק, לשכפל, לצלם, לתרגם, להקליט, לשדר, לקלוט ו/או לאחסן במאגר מידע בכל דרך ו/או אמצעי מכני, דיגיטלי, אופטי, מגנטי ו/או אחר – בחלק כלשהו מן המידע ו/או התמונות ו/או האיוורים ו/או כל תוכן אחר הכלולים ו/או שצורפו לספר זה, בין אם לשימוש פנימי או לשימוש מסחרי. כל שימוש החורג מציטוט קטעים קצרים במסגרת של ביקורת ספרותית אסור בהחלט, אלא ברשות מפורשת בכתב מהמוציא לאור.

לשם שטף הקריאה כתוב ספר זה בלשון זכר בלבד. ספר זה מיועד לגברים ונשים כאחד ואין בכוונתנו להפלות או לפגוע בציבור המשתמשים/ות.

(C) כל הזכויות שמורות

הוצאת הוד-עמי בע"מ

טלפון: 09-9564716

www.hod-ami.co.il

info@hod-ami.co.il

הודפס בישראל 2021

מסת"ב 978-965-361-426-0 ISBN

תוכן העניינים

13.....	פתח דבר.....
13.....	תוכנות ותקלות תוכנה.....
15.....	ייחודו של הספר.....
15.....	למי מיועד הספר.....
16.....	מחבר הספר וספרים קודמים שלו בתחום.....
17.....	חלק ראשון: מבוא.....
18.....	פרק 1: הבטחת איכות תוכנה – הגדרות ועקרונות.....
18.....	1.1 איכות תוכנה והבטחת איכות תוכנה – הגדרות.....
20.....	1.2 מוצר תוכנה מהו?.....
21.....	1.3 עקרונות הא"ת.....
21.....	1.4 שגיאות תוכנה, כשלי תוכנה ותקלות תוכנה.....
23.....	1.5 הגורמים לשגיאות תוכנה.....
27.....	1.6 הבטחת איכות תוכנה לעומת בקרת איכות תוכנה.....
28.....	1.7 הנדסת איכות תוכנה לעומת הנדסת תוכנה.....
28.....	ביבליוגרפיה נבחרת.....
30.....	פרק 2: גורמי איכות תוכנה.....
30.....	2.1 בעיות שהועלו במפגש החודשי של מועדון המחשוב – מיני אירוע כמבוא.....
31.....	2.2 הצורך בדרישות איכות תוכנה מקיפות.....
32.....	2.3 המודל הקלאסי של McCall לגורמי איכות תוכנה.....
33.....	2.3.1 גורמי איכות תוכנה במישור התפעול.....
35.....	2.3.2 גורמי איכות תוכנה במישור האחזקה.....
37.....	2.3.3 גורמי איכות תוכנה במישור המעבר.....
38.....	2.4 מודל ISO/IEC 25010 ומודלים אלטרנטיביים לגורמי איכות תוכנה.....
38.....	2.4.1 מודל ISO/IEC 25010.....
40.....	2.4.2 מודלים אלטרנטיביים של איכות תוכנה.....
43.....	2.5 הערכת תוכנה לפי גורמי איכות.....
45.....	ביבליוגרפיה נבחרת.....
47.....	פרק 3: אתגרים להבטחת איכות תוכנה.....
47.....	3.1 הקדמה.....
47.....	3.2 הייחודיות של הבטחת איכות תוכנה.....
50.....	3.3 האתגרים להא"ת בסביבת פיתוח התוכנה.....
54.....	ביבליוגרפיה נבחרת.....

פרק 4: הארגון של הבטחת איכות תוכנה.....55

4.1	מבוא	55
4.2	פעילויות הא"ת של ההנהלה	56
4.2.1	מדיניות איכות התוכנה	56
4.2	תפקידי ממונה ההנהלה על איכות התוכנה	58
4.3	תפקידי מנהל מחלקה האחראי ישירות לאיכות	59
4.4	נושאי איכות שבתפקידו של מנהל פרויקט	60
4.5	מערכת האיכות ויחידת הא"ת	60
4.5.1	מערכת הא"ת	60
4.5.2	יחידת הא"ת	61
4.5.3	המטלות של תחום א': יישום פעילויות הא"ת	62
4.5.4	המטלות של תחום ב': פעילויות הבטחת איכות העוסקות במוצר התוכנה	63
4.5.5	המטלות של תחום ג': פעילויות הבטחת איכות העוסקות בתהליך	63
4.6	הנלווים למערכת הא"ת	64
4.6.1	נאמני הא"ת ותפקידיהם	65
4.6.2	חברי ועדות הא"ת ותפקידיהם	65
4.6.3	פורומים של הא"ת – תפקידים ושיטות פעולה	66
	ביבליוגרפיה נבחרת	67

חלק שני: פעילויות ליישום תהליכי הא"ת.....69

פרק 5: הקמת תהליכי הא"ת והתיאום עם תהליכי פיתוח התוכנה.....70

5.1	הפעלת תהליכי הא"ת	70
5.2	תיאום תהליכי הא"ת עם תהליכי פיתוח התוכנה	71
	ביבליוגרפיה נבחרת	72

פרק 6: תהליך קדם-פרויקט – סקר חוזה.....73

6.1	מסיבת הסיום של פרויקט "שפיר" – מיני אירוע כמבוא	73
6.2	הקדמה	74
6.3	תהליך סקר חוזה ושלביו	75
6.4	מטרות הבדיקה והערכה בסקר חוזה	78
6.4.1	מטרות הבדיקה וההערכה בסקר טיוטת הצעה	78
6.4.2	מטרות הבדיקה וההערכה בסקר טיוטת חוזה	79
6.5	יישום סקר חוזה	80
6.5.1	מי מבצע את סקרי החוזה	80
6.5.2	ביצוע סקרי חוזה בפרויקטים גדולים	80
6.6	סקר חוזה בפרויקטים פנימיים	81
	נספח 6-א סקר טיוטת הצעה – רשימת תיוג	83
	נספח 6-ב סקר טיוטת חוזה – רשימת תיוג	86
	ביבליוגרפיה נבחרת	86

פרק 7: תוכנית הא"ת ותוכנית פרויקט.....87

7.1	מבוא	87
7.2	תהליך הכנת תוכנית הא"ת	88

89	7.3	המרכיבים של תוכנית הא"ת
89	7.3.1	מרכיבים של פעילויות יישום תהליכי הא"ת
90	7.3.2	מרכיבים של פעילויות הבטחת איכות המוצר
90	7.3.3	מרכיבים של פעילויות הבטחת איכות התהליכים
91	7.4	תהליך הכנת תוכנית פרויקט
92	7.5	שמוליק מודה למנהל המחלקה – מיני אירוע
93	7.6	המרכיבים של תוכנית פרויקט
93	7.6.1	מרכיבים של תוכנית הפיתוח
99	7.6.2	המרכיבים של תוכנית האיכות
102	7.7	התועלות של תוכנית פרויקט ותוכנית איכות
102	7.8	תוכניות פרויקט עבור פרויקטים קטנים ופרויקטים פנימיים
102	7.8.1	תוכניות פרויקט ותוכניות איכות עבור פרויקטים קטנים
104		ביבליוגרפיה נבחרת
106	פרק 8: עלות איכות תוכנה	
106		היקפן של עלויות איכות התוכנה – נתוני התעשייה
106	8.1	הפעם אושר התקציב – מיני אירוע כמבוא
108	8.2	המטרות של מדידת עלויות איכות תוכנה
109	8.3	המודל הקלאסי של עלויות איכות תוכנה
110	8.3.1	עלויות מניעה
111	8.3.2	עלויות הערכה
111	8.3.3	עלויות כשל פנימי
112	8.3.4	עלויות כשל חיצוני
113	8.4	היקפן של עלויות איכות התוכנה – נתוני התעשייה
114	8.5	המודל המורחב של עלויות איכות תוכנה
118	8.5.1	עלויות מניעה ניהולית
118	8.5.2	עלויות הערכה ניהולית
118	8.5.3	עלות כשל ניהולי פנימי
119	8.5.4	עלויות כשל ניהולי חיצוני
119	8.6	יישום של מערכת עלויות איכות תוכנה
119	8.6.1	הגדרת מודל של עלויות איכות תוכנה ופריטי עלות מתאימים
121	8.6.2	תכנון שטת איסוף נתוני עלות איכות התוכנה
121	8.6.3	יישום של מערכת עלויות איכות התוכנה שתוכנה
122	8.6.4	פעולות כתגובה לממצאי מערכת עלות איכות תוכנה
123	8.7	קשיים ביישום מדידות של עלויות איכות תוכנה
125		ביבליוגרפיה נבחרת
126	פרק 9: היעילות והעלות של תוכנית איכות לפרויקט פיתוח תוכנה – מודל הא"ת	
126	9.1	בסיס הנתונים של מודל הא"ת
129	9.2	מודל הא"ת
134	9.3	הפעלת מודל הא"ת להשוואת תוכנית איכות לפרויקט
137		ביבליוגרפיה נבחרת

פרק 10: רשומות איכות ובקרת תיעוד	138
10.1 הצרות של מנחם – מיני אירוע כמבוא	138
10.2 מבוא	139
10.3 המטרות של תהליכי בקרת תיעוד	141
10.4 יישום של בקרת תיעוד	141
10.4.1 הגדרה של רשימת הסוגים של מסמכים מבוקרים	141
10.4.2 תכנון ופיתוח של המסמכים המבוקרים	142
10.4.3 הגדרת השימושים הרגילים של המסמכים המבוקרים	143
10.4.4 עדכון של רשימת המסמכים המבוקרים	144
ביבליוגרפיה נבחרת	144
חלק שלישי: פעילויות להבטחת איכות מוצר התוכנה	145
פרק 11: הערכה של מוצר התוכנה לגבי התאמתו	146
11.1 מבוא	146
11.2 הערכה של תוכניות פרויקט לגבי התאמתן לחוזה	147
11.3 הערכה של מוצרי הפרויקט לגבי התאמתם לדרישות החוזה	147
11.4 הערכה של מוצרי הפרויקט לגבי סיכויי קבלתם על ידי הלקוח	148
11.5 התאמת מוצרי שלב תפעול מערכת התוכנה לגבי התאמתם לדרישות החוזה	149
11.6 הערכת של מוצר התוכנה על ידי מדידות	149
ביבליוגרפיה נבחרת	151
פרק 12: סקרי תיכון	152
סקר תיכון פורמלי	152
12.1 מבוא	152
12.2 סקר התיכון "שהצליח" – מיני אירוע כמבוא	154
12.3 סקר תיכון פורמלי (Formal Design Review – FDR)	155
12.3.1 המשתתפים בסקר תיכון פורמלי	155
12.3.2 ההכנות לסקר תיכון	157
12.3.3 ישיבת הסקר	158
12.4 סקרי עמיתים	160
12.4.1 המשתתפים בסקרי עמיתים	161
12.4.2 ההכנות לישיבת הסקר	162
12.4.3 התשתית לסקר עמיתים	163
12.4.4 ישיבת סקר העמיתים	163
12.4.5 פעילויות לאחר ישיבת הסקר – תיעוד	165
12.4.6 גרסאות של תהליך הביקורת	165
12.4.7 המועילות (האפקטיביות) והיעילות של סקרי עמיתים	166
12.4.8 מידת הכיסוי של סקרי עמיתים	169
12.4.9 השוואה של שיטות הסקר	170
12.5 חוות דעת של מומחים	171
ביבליוגרפיה נבחרת	171

פרק 13: מבחני תוכנה.....174

174	מבוא	13.1
178	יואל החליט לדלג על מבחני התוכנה שבתהליך – מיני אירוע כמבוא	13.2
179	אסטרטגיות של מבחני תוכנה	13.3
179	בחינה מצטברת לעומת בחינה בשיטת "המפץ הגדול"	13.3.1
180	סדר הביצוע בשיטת הבחינה המצטברת	13.3.2
182	בחינת "קופסה שחורה" או "קופסה לבנה" – החלטה אסטרטגית	13.3.3
189	מבחני תוכנה המונעים על ידי הדרישות	13.4
190	סוגי המבחנים של גורמי איכות התפעול	13.4.1
193	סוגי מבחנים של גורמי איכות האחזקה	13.4.2
195	סוגי מבחנים של גורמי המעבר	14.4.3
196	תכנון של תהליך הבחינה	13.5
196	מהם מקורות של מקרי בדיקה שישמשו במבחנים	13.5.1
200	מי צריך לבצע את המבחנים	13.5.2
200	היכן צריכים המבחנים להתקיים	13.5.3
200	תיעוד תכנון המבחנים	13.5.4
201	תכנון תהליך הבחינה	13.6
202	ביצוע תהליך המבחנים	13.7
202	תהליך היישום	13.7.1
204	תיעוד של תוצאות יישום המבחנים	13.7.2
205	מבחני תוכנה אוטומטיים	13.8
205	מבחנים אוטומטיים ומבחנים ידניים – השוואה	13.8.1
207	סוגים של מבחני תוכנה אוטומטיים	13.8.2
212	יתרונות וחסרונות של מבחנים אוטומטיים	13.8.3
213	תיקון שגיאות אוטומטי	13.9
215	אתרי אלפא ואתרי ביתא למבחני תוכנה	13.10
217	פעולות סקר תוכנה עבור שלבי התכנות והמבחנים	13.11
217	סקרי תוכנה (code reviews)	13.10.1
217	סקרי הסמכת תוכנה (software qualification reviews)	13.10.2
218	ביבליוגרפיה נבחרת	

פרק 14: הבטחת איכות של תפעול שירותי התוכנה.....222

222	מבוא	14.1
225	הצלחת "תוכנות אנוש בע"מ" – מיני אירוע כהקדמה	14.2
226	הבסיסים לאיכות גבוהה של שירותי תפעול	14.3
226	הבסיס הראשון – איכות חבילת התוכנה	14.3.1
227	הבסיס השני – מדיניות מהדורות וגרסאות תוכנה	14.3.2
229	היסוד השלישי – נהלי הבטחת איכות מיוחדים לשירותי התפעול	14.3.3
231	מודל הבגרות של אחזקת תוכנה – מודל לשלב התפעול	14.4
232	תהליכי ניהול הבטחת איכות של תפעול תוכנה	14.5
233	סקר החוזה של תפעול תוכנה	14.5.1
235	בקרת התקדמות שירותי התפעול	14.5.3
236	מדדי איכות לשירות תפעול תוכנה	14.5.4

237	14.5.5 עלות איכות שירותי תפעול התוכנה
240	ביבליוגרפיה נבחרת
242	פרק 15: מדדי איכות מוצר תוכנה
242	15.1 מהם מדדי איכות תוכנה? – מבוא
244	15.2 יישום של מדדי איכות תוכנה
244	15.2.1 הגדרת מדדי איכות
244	15.2.2 יישום המדדים
246	15.2.3 ניתוח נתוני המדדים על ידי הוועדה לפעולה מתקנת
246	15.2.4 פעולות בתגובה לניתוח תוצאות המדדים
248	15.3 מדדי מוצר התוכנה והסיווג שלהם
248	15.4 מדדי גודל מוצר התוכנה
251	15.5 מדדים של תכונות מוצר תוכנה
251	15.5.1 מדדים של פונקציונליות תוכנה
252	15.5.2 מדדי אמינות תוכנה
253	15.5.3 מדדי שימושיות תוכנה
254	15.5.4 מדדי יעילות התוכנה
255	15.5.5 מדדי אהזקתיות תוכנה
256	15.5.5 מדדי ניידות תוכנה
257	15.5.7 מדדי אפקטיביות תוכנה
258	15.5.8 מדדי פריזון תוכנה
258	15.5.9 מדדי בטיחות תוכנה
259	15.5.10 מדדי שביעות רצון מתוכנה
260	ביבליוגרפיה נבחרת
264	נספח 15A: יישום שיטת FSM
264	15A.1 שיטת הנקודות הפונקציונליות
267	15A.2 דוגמא – מערכת תוכנה לנוכחות עובדים
270	פרק 16: נהלים והוראות עבודה
270	16.1 נהלים והוראות עבודה – הגדרות
272	16.2 "מחשוב הלבואה" משלם 15,000 ש"ח פיצוי לרשת "סל-זול" על כשל מרכז התמיכה
273	16.3 נהלים – תכנים ומבנה
275	16.4 הוראות עבודה – תכנים ומבנה
276	16.5 הכנה, יישום ועדכון של נהלים והוראות עבודה
276	16.5.1 הכנת נהלים חדשים
277	16.5.2 יישום של נוהל חדש או נוהל מעודכן
277	16.5.3 עדכון נוהל
278	ביבליוגרפיה נבחרת
279	נספח 16-א: נוהל סקרי תיכון

חלק רביעי: פעילויות להבטחת איכות תהליכי תוכנה.....283

פרק 17: הערכת התאמה של התהליכים וסביבת הפיתוח.....284

17.1	מבוא	284
17.2	הערכת התאמתם של תהליכי מחזור החיים והתוכניות לדרישות	285
17.3	הערכת התאמתם של סביבת הפיתוח לדרישות	286
17.4	הערכת התאמתם לדרישות תהליכים של קבלני משנה ומשתתפים אחרים	286
17.5	הערכת התאמת תהליכי התוכנה לדרישות באמצעות מדידות	287
17.6	הערכת הידע והכישורים של הצוות, והתאמתם לדרישות הפרויקט	288
	ביבליוגרפיה נבחרת	288

פרק 18: תהליכי שיפור – פעולות מתקנות ומונעות.....289

18.1	צוות הפיתוח 7 נכשל בפרויקט "בני המפרץ" – מיני אירוע כהקדמה	289
18.2	הקדמה	291
18.3	תהליכי פעולות מתקנות ומונעות	293
18.3.1	איסוף מידע	293
18.3.2	ניתוח של המידע שנאסף	295
18.3.3	קביעת הגורמים לאי התאמה	295
18.3.4	פיתוח פתרונות	296
18.3.5	יישום של השיטות המשופרות	299
18.3.6	מעקב אחר פעולות מתקנות ומונעות – יישום ותוצאות	299
18.4	ארגון פעולות מתקנות ומונעות	300
	ביבליוגרפיה נבחרת	301

פרק 19: פעילויות הבטחת איכות תוכנה של משתתפים חיצוניים.....302

19.1	מבוא	302
19.2	מכרז פגאסוס – מיני אירוע	305
19.3	יתרונות וסיכונים של שילוב מבצעים חיצוניים	308
19.3.1	היתרונות העיקריים	308
19.3.2	הסיכונים העיקריים לקבלן הפרויקט	309
19.4	יתרונות וסיכונים של שימוש בתוכנה מוכנה	310
19.5	פעילויות להבטחת איכות תהליכים של מבצעים חיצוניים	311
19.6	פעולות הבטחת איכות להבטחת איכותה של תוכנה מוכנה	315
	ביבליוגרפיה נבחרת	318

פרק 20: מדדי איכות תהליך פיתוח תוכנה.....320

20.1	מבוא	320
20.2	הצפון נגד הדרום – מי זכה בסיבוב הנוכחי? – מיני אירוע כמבוא	322
20.3	מדדי תהליך פיתוח התוכנה	324
20.3.1	מדדי איכות תהליך פיתוח תוכנה	324
20.3.2	מדדי איכות פיתוח באמצעות תוכנה מוכנה	328
20.3.3	מדדי מועילות, תכנון ופירון בתהליך פיתוח תוכנה	330
20.3.4	מדדי עיבור מחדש בתהליכי פיתוח	333
20.4	מדדי תהליך תפעול התוכנה	333

334	20.4.1	מדדי העומס בתהליכי תפעול תוכנה
334	20.4.2	מדדי איכות לתהליכי תפעול התוכנה
335	20.4.3	מדדי אפקטיביות, תכנון ופריון של תהליכי תפעול תוכנה
336	20.5	מדדים לתהליכי אחזקת תוכנה
336	20.5.1	מדדי עומס תהליכי אחזקת תוכנה
338	20.5.2	מדדי איכות לתהליכי אחזקת תוכנה
338	20.5.3	מדדי אפקטיביות, תכנון ופריון בתהליכי אחזקת תוכנה
339	20.6	מדדי תהליך הניהול
339	20.6.1	מדדי ניהול התקדמות פרויקטים של פיתוח התוכנה
341	20.6.2	מדדי מיחזור בתהליכי פיתוח
341	20.7	המגבלות של מדדי איכות תוכנה
343		ביבליוגרפיה נבחרת
346	פרק 21:	תהליכי בקרת שינויים בתוכנה
346	21.1	מבוא
347	21.2	כיצד פרויקט מתוכנן היטב הפסיד מעל שני מיליון שקל
348	21.3	תהליך הטיפול בבקשה לשינוי בתוכנה
350	21.4	הוועדה לבקרת ביצוע שינויים ותפקידה בארגון
351	21.5	פעולות הא"ת המתייחסות לבקרת שינויים בתוכנה
351		ביבליוגרפיה נבחרת
352	פרק 22:	מיומנות וידע של הסגל – הדרכה והסמכה
352	22.1	מבוא
353	22.2	הפתעה לצוות הפיתוח של "שלושת האסים" – מיני אירוע כמבוא
354	22.3	מטרות ההדרכה
355	22.4	תהליך הדרכת סגל פיתוח התוכנה
359	22.5	פעילויות הדרכה לצוות יחידת הא"ת
360	22.6	המטרות של הסמכות לתפקיד
361	22.7	תהליך הפעלת הסמכות לתפקיד בארגון
365		ביבליוגרפיה נבחרת
367	חלק ה:	כלים ושיטות נוספים לסיוע לאיכות תוכנה
368	פרק 23:	תבניות ורשימות תיוג
368	23.1	מבוא
368	23.2	תבניות (templates)
370	23.3	מסגרת ארגונית ליישום תבניות
370	23.3.1	הכנת תבניות חדשות
371	23.3.2	יישום של תבניות
371	23.3.3	עדכון תבניות
371	23.4	רשימות תיוג (checklists)
373	23.5	המסגרת הארגונית ליישום רשימות תיוג
374	23.5.1	הכנת רשימות תיוג חדשות
374	23.5.2	עדכון רשימות תיוג

374.....	23.5.3 קידום השימוש ברשימות תיוג
374.....	ביבליוגרפיה נבחרת

פרק 24: ניהול תצורה.....375

375.....	24.1 מבוא
376.....	24.2 רכיבי תצורת תוכנה
378.....	24.3 שחרור מהדורות תצורת תוכנה
379.....	24.3.1 סוגים של שחרור תצורת תוכנה
380.....	24.3.2 מודלים להתפתחות מהדורות תצורת תוכנה
382.....	24.4 תיעוד של מהדורות תצורת תוכנה
383.....	24.5 תכנון ניהול תצורה
384.....	24.6 אספקת שירותי ניהול תצורת תוכנה
385.....	24.7 כלים ממוחשבים לביצוע מטלות ניהול תצורה
386.....	24.8 תפקיד ניהול תצורת תוכנה בארגון
386.....	24.9 פעילויות הא"ת הקשורות בניהול תצורת תוכנה
388.....	ביבליוגרפיה נבחרת

חלק ו: נספחים.....391

נספח א: סטנדרטים של תהליכי פיתוח תוכנה והא"ת.....392

392.....	1.א מבוא – סטנדרטים והשימוש בהם
395.....	2.א סטנדרט הא"ת IEEE 730-2014
397.....	3.א סטנדרט ISO/IEC 12207-2017 הנדסת מערכות ותוכנה – תהליכי מחזור החיים של תוכנה
397.....	3.1.א מטרות הסטנדרט
399.....	3.2.א תפישות
401.....	3.3.א סטנדרט 12207:2017 – תכנים
403.....	4.א סטנדרט IEEE 1012:2016 אימות ותיקוף מערכות ותוכנה
403.....	4.1.א מבוא
404.....	4.2.א תפישות סטנדרט IEEE 1012:2016
404.....	4.3.א סטנדרט IEEE 1012:2016 – תכנים
409.....	ביבליוגרפיה נבחרת

נספח ב: סטנדרטים ומודלים של ניהול איכות תוכנה.....411

411.....	1.ב אא תוכנות בע"מ – הפסד מיותר – מיני אירוע
413.....	2.ב התחום של סטנדרטים לניהול איכות
414.....	3.ב ISO/IEC/IEEE 90003
415.....	3.1.ב עקרונות מנחים לסטנדרטים ISO/IEC/IEEE 9001 ו-ISO/IEC/IEEE 90003
415.....	2.3.ב התפישות של ISO/IEC/IEEE 90003
416.....	3.3.ב התכנים של ISO/IEC/IEEE 90003
419.....	4.3.ב תהליך הסמכה ושמירת הסמכה לפי ISO/IEC/IEEE 90003

422	מודל בגרות היכולת המשולב – CMMI	4.ב
423	העקרונות של CMMI	4.1.ב
423	מבנה ותחומי תהליכים ב-CMMI	4.2.ב
427	תהליך הערכה באמצעות CMMI	4.3.ב
427	הניסיון שנרכש ביישומי CMM	4.4.ב
428	מודל אנשים של CMM לניהול משאבי אנוש (People CMM)	4.5.ב
430	פרויקט SPICE וסדרת סטנדרטים ISO/IEC33000 להערכת תהליך פיתוח תוכנה	5.ב
430	תפישות מאחורי המודלים של סדרת ISO/IEC 33000	5.1.ב
431	מבנה מודל ההערכה של סדרת ISO/IEC 33000	5.2.ב
434	תהליך הערכה לפי ISO/IEC 33000	5.3.ב
435	פרויקט SPICE	5.4.ב
436	ביבליוגרפיה נבחרת	
439	נספח ג: בקרת התקדמות פרויקטים	
439	מבוא	1.ג
441	לבסוף, פרויקט מוצלח – מיני אירוע	2.ג
443	המרכיבים של בקרת התקדמות פרויקטים	3.ג
445	בקרת התקדמות של פרויקטים פנימיים ומשתתפים חיצוניים	4.ג
446	יישום של בקרת התקדמות פרויקטים	5.ג
448	כלים ממוחשבים לבקרת התקדמות פרויקטים	6.ג
449	ביבליוגרפיה נבחרת	
451	נספח ד: AGILE תהליכים ופעולות הא"ת	
451	מבוא	1.ד
453	תהליך פיתוח אופייני לפי Agile	2.ד
454	תכונות נדרשות מאיש צוות Agile	3.ד
454	הבטחת איכות בפרויקטים של פיתוח תוכנה לפי Agile	4.ד
455	הניסיון שנרכש בפרויקטים במתודולוגיית Agile	5.ד
456	מגבלות של מתודולוגיות Agile	6.ד
456	ביבליוגרפיה נבחרת	
459	רשימת מונחים	
459	א-ב	
467	A-B	

פתח דבר

תוכנות ותקלות תוכנה

תוכנות מסוגים שונים משרתות אותנו כל יום, החל מהאינטרנט ועד לשליפת מזומנים מהכספומט. במידה גדולה וחיונית לא פחות, משרתות תוכנות את כל סוגי הארגונים החל מחשבונאות, ניהול מלאי, המשך בניהול קווי ייצור ועד לניווט טילים. התלות בשירותי התוכנות השונות היא גדולה ביותר עד קריטית.

האם אכן שירותי התוכנות האלה זמינים ותקינים תמיד? לצערנו התשובה היא לא. כמעט מדי יום אנחנו שומעים באמצעי התקשורת על תקלות תוכנה, וסובלים בעצמנו מתקלות תוכנה.

להלן כמה דוגמאות לידיעות על תקלות תוכנה שמצאנו לאחרונה בעיתונות:

מחדל חמור: אלפי תשובות לבדיקות קורונה מתעכבות

קריסת התוכנה האחראית לתוצאות בדיקות הקורונה גורמת לעיכוב גדול בתשובות לנבדקים. לדוגמא, ישנם נבדקים בבתי אבות שנדגמו לפני כשלושה ימים, ואולם עדיין לא קיבלו את התוצאות. במשרד הבריאות טוענים שקריסת המערכת היא תוצאה של העלייה במספר הבדיקות, שהמתכננים לא הביאו בחשבון.

מיקרוסופט מאשרת: קיימת בעיה חמורה ב-Windows 10

לפי ענקית התוכנה, הבאג במערכת ההפעלה של Windows 10 גורם לבעיות בזיהוי הקישוריות לאינטרנט, כאשר המערכת מציינת שאין חיבור אינטרנט זמין – למרות שישנו. מערכת ההפעלה סובלת מבעיות מיום היוולדה. המערכת מייצרת למשתמשים שלל צרות ותקלות – חלקן, על פי דעת מומחים, ממש אינן מחויבות המציאות. כעת מודה ענקית התוכנה בבאג נוסף במערכת.

תקלה של חברת NetApp גורמת לתקלות במערכות של ארגונים ברחבי ישראל

לא מעט ארגונים גדולים וחברות פרטיות בישראל סבלו היום מתקלות תוכנה חמורות שבסיסן במעבר לשעון חורף בישראל. מי שאחראית לתקלה היא חברת NetApp, שמפעילה שרתים וירטואליים רבים עבור מוסדות רבים בישראל, בהם אוניברסיטאות, משרד הבריאות וכן חברות פרטיות, כולם סבלו מבעיות מחשב חמורות. על פי דיווחים שונים. הארגונים חוו קשיים בתקשורת בין מחשבים, שמנעו מעובדים להתחבר במשך שעות ארוכות לשרתי המוסדות ולעבוד מרחוק.

שינוי שעון החורף טומן בחובו סיכונים לא מעטים לשירותי מיחשוב שונים שמסתמכים על NetApp בחישוב זמנים בטרנזקציות רבות.

תקלות – לא רק בישראל:

יותר מ-100 טיסות בוטלו בנמל התעופה Heathrow.

עקב תקלות מחשוב, בוטלו יותר מ-100 טיסות בנמל התעופה Heathrow. תקלות התוכנה פגעו במערכת קליטת הנוסעים ובתצוגת לוחות ההמראות בשדה.

כדאי לציין שתקלת תוכנה דומה קרתה, רק לפני חודשים אחדים, במערכת קליטת הנוסעים ותצוגת לוחות המראת המטוסים של חברת British Airways בנמל התעופה Heathrow. אותה תקלה גרמה לביטול יותר מ-100 טיסות, ולדחיית ההמראה של 200 טיסות אחרות.

תקלות תוכנה חמורות במטוסי Boeing 737 MAX

מטוסי Boeing 737 MAX קורקעו והושבתו בהחלטת FAA, רשות התעופה הפדרלית של ארה"ב, לאחר אסונות ההתרסקות של מטוס חברת Lion ושל מטוס Ethiopian Airlines. מספר הקורבנות הגיע ל-346. החלטת ה-FAA התקבלה לאור תקלות תוכנה חמורות בתוכנת מערכות המטוס, כולל סימולטור הטיסה של המטוס, המשמש לתרגול הטייסים.

רק אחרי יותר משנה וחצי של השבתת המטוסים, החליט ה-FAA על מתן אישור מחודש לכשירות טייס למטוס Boeing 737 Max. תקופת השבתה שימשה לתיקון ליקויי התוכנה של מערכות המטוס והסימולטור, ולביצוע מבחני מערכת מקיפים לתוכנת מערכות המטוס השונות.

רק לאחר קבלת אישור כשירות לטיס הגיעה חברת Boeing להסדר התביעה של FAA: תשלום קנס בסך 243 מיליוני דולר, תשלום פיצויים לקורבנות של התרסקות המטוסים בסך 500 מיליוני דולר, ותשלום פיצוי לנזקי חברות התעופה בסך 1770 מיליוני דולר. סכומים אלה הם, למעשה, רק חלק מהנזקים שנגרמו לחברת Boeing עקב ליקויי התוכנה במטוס.

אלפי מכוניות Tesla ננעלו בלי אפשרות להיכנס לרכב.

אלפי מכוניות Tesla בארה"ב ננעלו למשך שעות ארוכות, מבלי אפשרות לבעלים להיכנס למכונית. כל זאת בגלל תקלה בעדכון תוכנה פשוט באפליקציה סלולרית, המאפשרת לבעל הרכב לתקשר עם רכבו. התקלה גרמה לנתק מוחלט של הבעלים מהרכב, ובנוסף לחוסר אפשרות של בעל הרכב לתקשר עם החברה.

כל התקלות הללו הן רק דוגמאות לתקלות תוכנה הנגרמות כל יום. תקלות תוכנה אלה גורמות נזקים מסוגים שונים ובהיקפים שונים. המשותף לכל התקלות הוא שאפשר היה למנוע אותן אילו מפתחי התוכנה היו מבצעים כהלכה את תהליכי הבטחת איכות תוכנה הנדרשים: בתיאום, במעקב ובהערכה.

המכלול הרחב של תהליכי הבטחת איכות תוכנה לאיתור תקלות תוכנה עוד בתהליך הפיתוח, ומניעת היווצרות תקלות, הם תוכנו של ספר זה.

ייחודו של הספר

ייחודו של הספר הוא בנושאים הבאים:

- הדגשה מיוחדת לשלב היישום של תהליכי הבטחת איכות תוכנה. זאת על ידי ניתוח אירועים והסקת מסקנות, דוגמאות והערות ליישום המשובצות בטקסט.
- התייחסות נרחבת לנושאים כמותיים בהבטחת איכות תוכנה: עלויות איכות, כולל מודל מורחב לעלויות איכות תוכנה (שפותח על ידי המחבר), ומדדי איכות למוצר התוכנה ולתהליך הפיתוח.
- דיון בהבטחת איכות בפיתוח עצמי של תוכנה – פיתוח תוכנה על ידי יחידה בארגון לשימוש יחידות אחרות בארגון.
- הצגת הבעיות בפרויקטים בהשתתפות גורמי חוץ בפיתוח התוכנה: קבלני משנה (מיקור חוץ) וספקי תוכנה מוכנה, ומתן פתרונות. זאת בפרק מיוחד, בסעיפים ובקטעים בפרקים.
- הקדשת פרק מיוחד לנושא הבטחת איכות בתהליכי תפעול תוכנה שוטף ואחזקת תוכנה.
- דיון רחב בתקנים בינלאומיים של איכות תוכנה, וביניהם תקן הבטחת איכות תוכנה IEEE 730.
- הצגת נושא הארגון של הבטחת איכות: פעילויות ההנהלה בתחום האיכות, תפקידי המנהלים בהבטחת איכות התוכנה ותפקידי מנהל הפרויקט. חלק מרכזי בפרק עוסק ביחידה להבטחת איכות התוכנה, מבנה היחידה והמטלות שלה.
- הקדשת פרק להצגת מודל היעילות והעלות של תוכנית הבטחת איכות לפרויקט פיתוח תוכנה. המודל (שפיתח המחבר) מחשב את יעילות תוכנית הבטחת האיכות במונחים של (א) אחוז השגיאות הנותרות במוצר התוכנה בסיומו של תהליך הפיתוח, ו-(ב) העלות הכוללת של תהליכי הבטחת איכות התוכנה שתוכננו. בפרק מובאת השוואה, באמצעות המודל, של שתי תוכניות הבטחת איכות לפרויקט.

למי מיועד הספר

הספר מיועד לכמה אוכלוסיות של קוראים:

- סטודנטים בהנדסת תוכנה ומדעי המחשב באוניברסיטאות ובמכללות
- מקצוענים העוסקים בפיתוח תוכנה, שנתקלים בנושאי איכות, במהלך עבודתם בפרויקטים של פיתוח ואחזקת תוכנה, כולל מקצוענים העוסקים בהבטחת איכות תוכנה.
- מנהלים של יחידות פיתוח תוכנה בבתי תוכנה ובארגונים השונים.
- תלמידי בתי ספר להנדסאים בתחום התוכנה ותלמידים במסגרות הכשרה מקצועית באותו תחום.
- קוראים המעוניינים בהסמכה כ"מהנדס איכות תוכנה מוסמך" (CSQE) של ASQ, או בהסמכה כ"אנליסט איכות תוכנה מוסמך" (CSQA) של QAI.
- וכמובן – לומדים בקורסים של "הבטחת איכות תוכנה".

מחבר הספר וספרים קודמים שלו בתחום

מחבר הספר, ד"ר דניאל גלין, הוא בעל תואר DSc מהפקולטה להנדסת תעשייה וניהול של הטכניון. הידע שלו בתחום הבטחת איכות תוכנה נרכש בשנים רבות של ייעוץ, הוראה וקתיבה בתחום. כמרצה בכיר בטכניון, במרכז האקדמי רופין ובמקומות נוספים, הרצה בנושאי הבטחת איכות תוכנה, ניתוח ועיצוב מערכות מידע ומערכות מידע תחרותיות-אסטרטגיות. ד"ר גלין כתב ספרים ומאמרים רבים בתחומי התמחותו.

הספר הנוכחי הוא הרביעי שכתב ד"ר גלין בתחום הבטחת איכות תוכנה. קדמו לו:

- "הבטחת איכות תוכנה" (בשיתוף עם ד"ר זיגמונד בלובבנד) שהופיע בהוצאת "אופוס" בשנת 1995.
- "Software Quality Assurance – From Theory to Implementation", בהוצאת Addison-Wesley, בריטניה, 2004.
- "Software Quality – Concepts and Practice" בהוצאת IEEE Press Computer Society – Wiley, ארה"ב, 2017.

הספר הנוכחי, "הבטחת איכות תוכנה – מעקרונות ליישום", עוקב אחר הספר הקודם במבנה ובתכנים, תוך עדכונים ושינויים.

חלק ראשון:

מבוא

בחלק זה, הפותח את הספר, יוצגו הגדרות ונושאי רקע הקשורים באיכות תוכנה:

- ⊙ הבטחת איכות תוכנה – הגדרות ועקרונות
- ⊙ גורמי איכות תוכנה
- ⊙ אתגרים להבטחת איכות תוכנה
- ⊙ ארגון לצורך הבטחת איכות תוכנה
- ⊙ העולם של הבטחת איכות תוכנה – סקירה כללית

פרק 1:

הבטחת איכות תוכנה – הגדרות ועקרונות

בפרק זה:

- ⊙ איכות תוכנה והבטחת איכות תוכנה – הגדרות
- ⊙ מוצר תוכנה מהו?
- ⊙ עקרונות הבטחת איכות תוכנה
- ⊙ שגיאות תוכנה, כשלי תוכנה ותקלות תוכנה
- ⊙ הגורמים לשגיאות תוכנה
- ⊙ הבטחת איכות תוכנה לעומת בקרת איכות תוכנה
- ⊙ הנדסת איכות תוכנה לעומת הנדסת תוכנה

1.1 איכות תוכנה והבטחת איכות תוכנה – הגדרות

בסעיף זה נציג את ההגדרות הבסיסיות הבאות:

- איכות תוכנה
- הבטחת איכות תוכנה (הא"ת)
- הא"ת – הגדרה מורחבת
- המטרות של הא"ת

איכות תוכנה – הגדרה

מקור: IEEE Std. 730-2014

איכות תוכנה היא המידה שבה מוצר התוכנה ממלא את הדרישות שנקבעו, כאשר הדרישות שנקבעו מייצגות את צורכי בעלי העניין ואת רצונותיהם וציפיותיהם.

ההגדרה מציגה שני צדדים של איכות התוכנה. מצד אחד דורשת מילוי בדרישות שנקבעו ומהצד השני מתייחסת לשביעות הרצון של בעלי העניין והמשתמשים. תוכנה בעלת איכות גבוהה נדרשת

למלא במלואן את כל הדרישות הכתובות, בין אם הוגדרו לפני תחילת הפיתוח או במהלך הפיתוח, ובנוסף למלא את התקנים הרלוונטיים והמוסכמות המקצועיות. שביעות הרצון של בעלי העניין והמשתמשים מושגת על ידי מילוי צורכיהם ורצונותיהם.

הא"ת – הגדרה

ההגדרה שהוצעה על ידי IEEE, והיא אחת הנהוגות ביותר:

הא"ת היא מערכת פעילויות המגדירות ומעריכות את התאמת תהליך פיתוח התוכנה, מתעדות ומבטיחות, ברמת מהימנות נאותה, שתהליך פיתוח התוכנה הוא תקין ומייצר תוכנה באיכות נאותה לשימושים המיועדים. תכונה עיקרית של הא"ת היא האובייקטיביות של פעולות הא"ת ביחס לפרויקט. פעולות הא"ת יכולות להיות עצמאיות מהפרויקט מבחינה ארגונית, כלומר חופשיות מלחצים טכניים, ניהוליים וכלכליים מצד הפרויקט.

ההגדרה מתאפיינת בנושאים הבאים:

- תכנון ויישום שיטתי. הא"ת מבוססת על תכנון ויישום של פעולות המשולבות בשלבים השונים של תהליך פיתוח התוכנה. פעולות אלה מיועדות להבטיח ללקוח שמוצר התוכנה ימלא את כל הדרישות הטכניות שהוגדרו.
- ההגדרה מתייחסת להתאמת מוצר התוכנה לדרישות הטכניות ולהתאמתו לשימוש המתוכנן על ידי בעלי העניין.
- ההגדרה אינה מתייחסת לתכונות חשובות של תהליך הפיתוח, שמירת לוח הזמנים ותקציב הפרויקט.
- ההגדרה אינה כוללת התייחסות לתפעול השוטף של מערכת התוכנה ולתחזוקתה.

הא"ת – הגדרה מורחבת

ההגדרה המורחבת יוצרת התייחסות לחשיבות של תהליך התפעול השוטף של התוכנה ולהשפעה החשובה של שמירת תקציב ולוח זמנים של תהליך הפיתוח על איכות מוצר התוכנה.

הא"ת היא מערכת פעילויות המגדירות ומעריכות את התאמת תהליך פיתוח התוכנה, מתעדות ומבטיחות ברמת מהימנות נאותה שתהליך פיתוח התוכנה הוא תקין ומייצר תוכנה ברמת איכות נאותה לשימושים המיועדים, לביצוע שירותי התוכנה המיועדים, וממלא את דרישות לוח הזמנים והתקציב של הפרויקט.

המטרות של הא"ת

המטרות של הא"ת הן:

- להבטיח שמוצר התוכנה ושירותי התוכנה יהיו ברמה מקובלת של איכות כשהם תואמים את הדרישות הטכניות והפונקציונליות ובעלי איכות מתאימה לשימושים המיועדים לתוכנה.
- בהתאם להגדרה המורחבת של הא"ת, להבטיח ברמה מקובלת של מהימנות שתהליך פיתוח התוכנה ושירותי התוכנה יעמדו בדרישות התקציב ולוח הזמנים.

- ליזום ולנהל פעולות לשפר ולהגדיל את יעילות פיתוח התוכנה, שירותי התוכנה ופעולות הא"ת. פעולות אלו מניבות שיפורים בהשגת הדרישות הפונקציונליות והניהוליות, תוך הפחתת עלויות הפיתוח.

דרכים לעודד הבטחת איכות בארגונים גדולים מוצעים על ידי (Poeth and Heinemann 2018).

1.2 מוצר תוכנה מהו?

כאשר אנחנו חושבים על מוצר תוכנה, צץ בדמיונו אוסף של פקודות תכנות שמתייחסים אליו כ"קוד". לאמיתו של דבר, למוצר תוכנה מקצועי, ה"קוד" לכשעצמו אינו מספיק. מוצר התוכנה צריך לעמוד בתיקוני טעויות במהלך הפיתוח וגם בשירותי אחזקה של תיקוני טעויות, התאמות ושיפורים בתקופת התפעול השוטף של התוכנה. לשם כך מכיל מוצר התוכנה מרכיבים שונים המיועדים להבטיח הצלחה של תהליך הפיתוח ושל השירותים המסופקים על ידי התוכנה. ההגדרה של ISO/IEC/IEEE מפרטת את המרכיבים האלה.

הגדרת מוצר תוכנה המקור: ISO/IEC/IEEE Std.90003/2018

מרכיבי מוצר התוכנה הם:

תוכנה. תוכניות המחשב המפעילות את המחשב כדי לבצע את היישומים הדרושים. קיימים סוגים שונים של תוכנה, כמו תוכנת מקור, תוכנת הרצה (executable code), תוכנת ניסוי ועוד.

נהלים. נהלים מגדירים את הדרך ולוח הזמנים שעל פיהם תוכניות מחשב מבוצעות, את תהליך הטיפול בתקלות שכיחות וכדומה.

תיעוד. המטרה של התיעוד היא להדריך ולתמוך במפתחים של גרסאות תוכנה חדשות, בצוותי אחזקה ובמשתמשים. התיעוד כולל דוחות תכנון (תכנון כולל תכנון כללי – ניתוח (analysis) וגם תכנון מפורט – תיכון, עיצוב (design)), דוחות ניסוי, מדריכים למשתמש ומדריכים למתכנת ועוד.

נתונים. נתונים דרושים להפעלת התוכנה דרושים נתונים כמו רשימות קודים ופרמטרים. בנוסף משתמשים בנתוני ניסוי סטנדרטיים לבדיקת תקינות התוכנה לאחר שבוצעו שינויים, תיקונים או הרחבות תוכנה. הרצת המערכת עם נתוני הניסוי יכולה לסייע גם באיתור הגורמים לתקלה.

לסיכום, מוצעת כאן ההגדרה הבאה למוצר התוכנה:

מוצר תוכנה הוא צירוף של מרכיבים הכרחיים כדי להבטיח תפקוד תקין ואחזקה יעילה במהלך מחזור החיים של התוכנה. המרכיבים הם: (1) תוכניות מחשב ("קוד"), (2) נהלים, (3) תיעוד, (4) נתונים לצורך התפעול השוטף והאחזקה.

כאן המקום לציין כי הא"ת עוסקת בכל מרכיבי מוצר התוכנה. הרכב מרכיבי מוצר התוכנה משתנה במידה ניכרת לפי כלי הפיתוח והמתודולוגיה.

1.3 עקרונות הא"ת

העקרונות שיוצגו כאן הם בעקבות תקן ISO 9000:2000.

העקרונות הבאים מנחים את הארגון בתהליכים המבטיחים שמוצרי התוכנה ושירותי התוכנה ימלאו את צורכי בעלי העניין ורצונותיהם.

- **התמקדות בלקוחות.** הארגונים תלויים בלקוחותיהם, ולפיכך הם צריכים להבין את צורכיהם הנוכחיים והעתידיים, למלא את דרישותיהם ולהשיג את שביעות רצונם.
- **מנהיגות.** מנהיגי הארגון צריכים ליצור סביבה פנימית שבה המועסקים מעורבים בהשגת מטרות האיכות.
- **מעורבות אנשים-מועסקים.** המעורבות של המועסקים מכל רמות הארגון מאפשרת לארגון להנות מהיכולות שלהם לקדם את נושאי הבטחת איכות.
- **גישה תהליכית.** התוצאה של ניהול פעילויות ומשאבים כתהליכים היא שיפור היעילות.
- **גישה מערכתית בניהול.** ניהול תהליכים משיג אפקטיביות ויעילות גבוהות יותר על ידי זיהוי, ניתוח והבנה של תהליכים מקושרים.
- **שיפור מתמשך.** שיפור מתמשך ומשולב של איכות, יעילות ואפקטיביות של ביצועים הם יעד קבוע של הארגון.
- **גישה עובדתית לקבלת החלטות.** החלטות צריכות להתבסס על נתונים ומידע.
- **קשרי ספקים הדדיים ותועלתיים.** הבנה שיחסי הארגון עם הספקים, המבוססים על תועלות הדדיות תורמות לביצועים משופרים של הארגון ביחס לאיכות, יעילות ואפקטיביות.

1.4 שגיאות תוכנה, כשלי תוכנה ותקלות תוכנה

כדי להבין טוב יותר את מצבי חוסר האיכות בתוכנה אנו מבחינים בין שגיאות תוכנה, כשלי תוכנה ותקלות תוכנה.

שגיאת תוכנה (software error). שגיאות תוכנה שהן המקור לתקלות תוכנה, נוצרות בידי מתכנני תוכנה ומתכנתים. שגיאת תוכנה יכולה להיות דקדוקית באחת או יותר משורות התוכנה או שגיאה לוגית בביצוע אחת או יותר מהדרישות שנקבעו

כשל תוכנה (software fault). כשל תוכנה הוא טעות תוכנה הגורמת לתפקוד לא תקין של התוכנה במצב תפעולי מסוים. יש לציין שלא כל שגיאות התוכנה נעשות לכשלי תוכנה. במקרים רבים שורות תוכנה שגויות לא משפיעות על התפקוד של התוכנה (כלומר לא נעשות לכשלי תוכנה). כדאי לציין כאן מקרים שבהם שגיאת תוכנה מתוקנת על ידי שגיאת תוכנה נוספת "המנטרלת" את השגיאה הקודמת, וכתוצאה אין כשל תוכנה.

תקלת תוכנה (software failure). באופן טבעי אנחנו מתעניינים בתקלות תוכנה המפסיקות/מפריעות לתפעול התקין של התוכנה. תקלת תוכנה נגרמת כאשר נוצר מצב הפעלה שבו כשל תוכנה

מופעל, ותפקוד התוכנה סוטה מהנדרש. במקרים רבים כשל תוכנה לעולם אינו מופעל, זאת כתוצאה מכך שלמשתמש לא היה עניין ביישום המסוים שבו מצוי כשל התוכנה.

להדגמת המצבים השונים של חוסר איכות בתוכנה מובאות הדוגמאות הבאות:

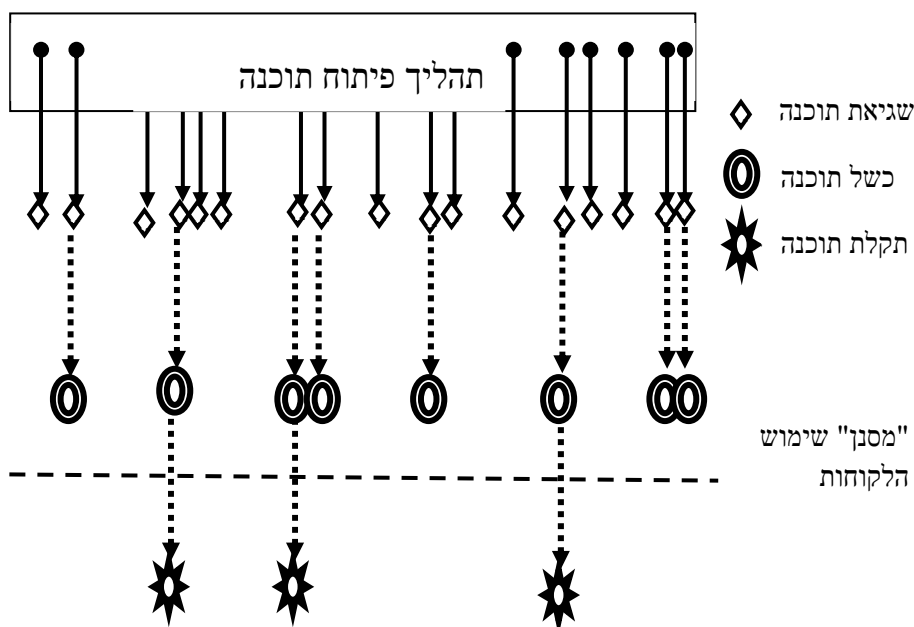
(1) **מערכת שכר שעות נוספות בבית ספר.** על פי הסכם העבודה: א. עבור שתי שעות ההוראה הנוספות הראשונות – תוספת שכר של 50%. ב. עבור שעות הוראה מעבר לשתי השעות הנוספות הראשונות – תוספת שכר של 100%. מפתחי התוכנה שגו כשקבעו תוספת שכר של 50% עבור כל שעת הוראה נוספת. כאן בבירור יש **כשל תוכנה** חמור. במשך חמש שנים לא הייתה תקלה כלשהי בחישוב תוספת השכר עבור השעות הנוספות. הסיבה שכשל התוכנה לא הפך לתקלת תוכנה נעוצה בהוראת המנהל שאינה מתירה הוראה של יותר משתי שעות הוראה נוספות

(2) **תחנה מטאורולוגית ממוחשבת.** התחנה נועדה לרשום ולהעביר בתקשורת אחת לחמש דקות את נתוני הטמפרטורה ומהירות הרוח. מפתחי המערכת נדרשו להפעיל מנגנון הגנה על התחנה ולהפסיק את פעילותה כאשר מהירות הרוח עולה על 150 קשר או שהטמפרטורה עולה על 50 מעלות. שגיאת המתכנת הייתה שקבע את טמפרטורת ההגנה כ-150 מעלות. כאן המתכנת שגה וגרם **לכשל תוכנה**. החברה הצליחה למכור את התחנות מתוצרתה למדינות צפון אירופאיות והן פעלו בהצלחה ללא כל תקלת תוכנה. כשל התוכנה התגלה רק לאחר 3 שנים כשהחברה המייצרת את התחנה מכרה אותה להתקנה במדינה מרכז אפריקנית. כאן מערכת ההגנה לא הופעלה בתנאים המדבריים החמורים וציווד התחנה נפגע. כאן הופעל **כשל התוכנה** ונוצרה **תקלת תוכנה**, ובמקרה זה גם נזק לציווד.

(3) **תצוגת טמפרטורות בערי העולם.** מערכת התוכנה נועדה לאסוף נתוני טמפרטורה עכשוויות מ-30 ערים ברחבי העולם ולהציג את התוצאות שהתעדכנו אחת לעשר דקות. לתצוגה תוכנו שתי גרסאות: גרסה עברית וגרסה אנגלית. המערכת תוכננה ותוכנתה היטב אך בתכנות גרסת התצוגה האנגלית שגה המתכנת ורשם Landon במקום London וגם Kapetown במקום Cape town. כאן בבירור **שגיאות תוכנה** שאינן משפיעות על תפקוד המערכת (אך כמובן שדורשת תיקון).

איור 1.1 מציג את היחסים בין שגיאות תוכנה, כשלי תוכנה ותקלות תוכנה. מתוך 17 שגיאות התוכנה המוצגות באיור רק 8 נעשות לכשלי תוכנה ומתוכן רק 3 הופכים לתקלות תוכנה. מאפייני השימוש בתוכנה על ידי הלקוח שקובע אילו יישומים יופעלו ובדרך זאת קובע מי מכשלי התוכנה יהפכו לתקלות תוכנה. במילים אחרות ניתן לראות את מאפייני השימוש בתוכנה כ"מסנן" (פילטר) לתקלות תוכנה – "מסנן" שימושי הלקוח.

איור 1.1: שגיאות תוכנה, כשלי תוכנה ותקלות תוכנה



שיטות למיון שגיאות תוכנה מוצגות על ידי Hernandez-Gonzalez and Lozano (2018) וגם Balogun and Aro T.O. (2019).

1.5 הגורמים לשגיאות תוכנה

שגיאות תוכנה הן המקור לכשלי תוכנה ולתקלות תוכנה. נבחן עתה מהם הגורמים לשגיאות תוכנה, ידע זה יוכל לסייע במניעת שגיאות תוכנה. יוקדם ויאמר – שגיאות תוכנה כולן שגיאות אנוש, אשר נעשו על ידי מנתחי מערכות, מתכנתים, בוחני תוכנה, מומחי תיעוד מנהלים ולפעמים גם לקוחות ונציגיהם. גם במקרים שאפשר לייחס שגיאה לגורם ממוחשב כמו בוחן תוכנה אוטומטי, הרי שגם במקרה זה גורם אנושי היה זה שגרם לתקלה בבוחן התוכנה במהלך פיתוח הבוחן. סיווג הגורמים לשגיאות תוכנה לתשע קבוצות מוצג כאן.

א. הגדרת דרישות שגויה

הגדרת דרישות לקויה, אשר ברוב המקרים הוכנה על ידי הלקוח, היא אחד הגורמים העיקריים לשגיאות תוכנה. הסוגים האופייניים והשכיחים לשגיאות מסוג זה הם:

- הגדרת דרישות לקויה.
- חוסר בדרישות עיקריות.
- הגדרת דרישות בלתי שלמה.

לדוגמא: רשימת ההנחות בתוכנה לחישוב הארנונה העירונית. תקנות הארנונה כוללות רשימה ארוכה למדי של הנחות: לאזרחים ותיקים, לעולים חדשים, למשפחות חד-הוריות ועוד. ברשימה שנמסרה לצוות הפרויקט נשמטה הנחה לסטודנטים, אף שאושרה על ידי מועצת העיר.

- הוספת דרישות שאינן דרושות, תפקודים שלא צפויים ליישום.

ב. כשלי תקשורת בין המפתחים ללקוח

גורם שגיאאות זה נפוץ בכל השלבים של תהליך הפיתוח.

- אי הבנה של דרישות הלקוח המופיעות במסמך הדרישות.
- אי הבנה של דרישות הלקוח לשינויים במהלך הפיתוח, כאשר דרישות השינוי מוצגות בכתב או בעל פה.
- אי הבנה של תשובות הלקוח לשאלות המפתחים בנושאי הפרויקט.
- חוסר תשומת לב להודעות הלקוח המתייחסות לשינויים בדרישות.

ג. סטיות במודע של המפתחים מדרישות הפרויקט

בנסיבות מסוימות, עשויים מפתחים לסטות ביוזעין מהדרישות המוגדרות לפרויקט, סטייה שלעיתים קרובות גורמת לשגיאות. הנסיבות העיקריות המביאות לסטיות ביוזעין מהדרישות הן:

- רצון המפתח למחזר חלקים מפרויקט תוכנה קודם ללא ניתוח מספיק של השינויים שיידרשו כדי למלא את דרישות הפרויקט הנוכחי.
- המפתח מחליט להשמיט חלקים מהיישומים הנכללים בפרויקט, בניסיון לעמוד טוב יותר בדרישות לוח הזמנים והתקציב.
- המפתח יוזם שיפורים בפרויקט מבלי לקבל אישור הנהלה או הלקוח. שיפורים מסוג זה, לעיתים קרובות, מתעלמים מדרישות הפרויקט הנחשבות "משניות" על ידי המפתח.

ד. שגיאות לוגיות בתכנון

שגיאות לוגיות בתכנון יכולות להיגרם על ידי המקצוענים השונים המעורבים בשלבי התכנון של הפרויקט החל ממנתחי מערכות הפרויקט ועד מהנדסי תוכנה, המעצבים את הגדרות התכנון מתוך מסמך הדרישות. שגיאות לוגיות אופייניות כוללות:

- שימוש באלגוריתם שגוי לייצוג דרישות המערכת.
- הגדרת תהליכים הכוללים שגיאות ברצף התהליך.

לדוגמא: במערכת ניהול הגבייה **הדרישה** – לקוח שחובו 5000 ש"ח או יותר יקבל התראה כתובה שתופק על ידי המערכת באופן אוטומטי. במידה ובתום חודש ממשלוח ההתראה לא שילם את חובו או הקטין אותו מתחת ל-5000 ש"ח, יועבר להחלטתו של מנהל השיווק שיחליט האם להעבירו לטיפול המחלקה המשפטית או שלא לטפל בגבייה בשלב זה. **מנתח המערכת** רשם – לקוח שחובו עולה על 5000 ש"ח יקבל התראה כתובה שתופק על ידי המערכת באופן אוטומטי. במידה ובתום חודש ממשלוח ההתראה לא שילם את חובו או הקטין אותו מתחת ל-5000 ש"ח, יועבר חוב הלקוח לטיפול המחלקה המשפטית.

- הגדרה שגויה של תנאי גבול.

דוגמא: בהנחות של הארנונה העירונית **הדרישה** – הנחה של 30% תינתן למשפחות בנות יותר מ-4 ילדים. **מנתח המערכת** רשם – הנחה של 30% תינתן למשפחות בנות 4 ילדים ויותר.

- חוסר הגדרה של תגובה נדרשת למצב לא חוקי.
- במערכת ממוחשבת למכירת כרטיסי תיאטרון **הדרישה** להגביל את המכירה ללקוח ל-10 כרטיסים לאירוע. **מנתח המערכת** – כלל הודעה ללקוח על המגבלה הכמותית אך שכה לכלול תגובת מערכת כאשר הלקוח ביקש להזמין מעל 10 כרטיסים לאירוע.

ה. שגיאות תכנות

יש מגוון גדול של סיבות הגורמות למתכנת לעשות שגיאות תכנות. סיבות אלה כוללות הבנה לקויה של מסמכי התכנון, שגיאות לשוניות בשפת התכנות, שגיאות בהפעלת כלי הפיתוח, שגיאות בבחירת הנתונים המתאים במסד הנתונים ועוד.

ו. אי קיום הוראות התכנות והתייעוד

כמעט כל יחידת פיתוח תוכנה מאמצת לעצמה סטנדרטים והוראות שמגדירים תוכן, סדר וצורה לתכנות ולתייעוד המפותחים ביחידה. היחידה מפתחת לצורך זה הוראות פיתוח, תבניות (templates) ומסמכים אחרים. צוותי הפיתוח נדרשים למלא את הסטנדרטים וההוראות.

מכיוון שאפשר להניח כי אי מילוי דרישות הסטנדרט או ההוראות, בדרך כלל, אינו יוצר שגיאת תוכנה, נשאלת השאלה מדוע אי מילוי הסטנדרטים וההוראות נחשב לגורם לשגיאות תוכנה. התשובה קשורה בטיפול העיתי במערכת התוכנה, כאשר תוכנה שאינה על פי הסטנדרטים וההוראות גורמת קשיים בהבנת המערכת בעתיד: לאנשי הפיתוח הממשיכים בפיתוח המערכת ולאנשי האחזקה שיפעילו את המערכת. קשיים אלה יכולים לגרום לשגיאות תוכנה רבות יותר. פירוט נסיבות שיגרמו לקשיים:

- אנשי צוות שיידרשו לתאם את התכנות שהם מבצעים עם חבר צוות ש"אינו מציית", צפויים ליותר קשיים מהרגיל.
- אנשי צוות המחליפים איש צוות ש"אינו מציית" (שפרש או קודם לתפקיד אחר).
- צוות סקר התכנון שאמרו לבדוק את התכנון יתקשה לבדוק מסמכים שנכתבו שלא על פי הסטנדרט וההוראות.
- צוות מבחני התוכנה יתקשה לבחון חלק תוכנה שנכתבו שלא על פי הסטנדרט וההוראות.
- צוות האחזקה יתקשה בתיקון "באגים" שהתגלו על ידי המשתמשים, בהתאמת התוכנה למשתמשים חדשים ובהוספת יישומים משלימים.

ז. ליקויים בתהליך הבחינה

- ליקויים בתהליך הבחינה יכולים לגרום לשגיאות שלא התגלו או שלא תוקנו. סיבות הליקוי:
- תוכנית מבחני תוכנה שאינה שלמה, כלומר חלקי מערכת התוכנה אינם נכללים בתוכנית המבחנים.
 - תקלות בתייעוד ובטיפול של שגיאות תוכנה שאותרו.
 - תקלות בתיקון של שגיאות תוכנה שאותרו, כתוצאה מזיהוי שגוי של גורם השגיאה.

- בחינה לא מלאה של תיקוני תוכנה שבוצעו.
- תיקון של רק חלק מהשגיאות שאותרו, וזאת כתוצאה מרשלנות או לחץ זמן.

ה. שגיאות בממשק המשתמש ובנהלים

שגיאות בממשק המשתמש (user interface) מדריכות את המשתמש באופן שגוי בביצוע פעולות שונות ובעיקר בפעולות קלט ופלט ועיבוד נתונים. שגיאות בנהלים מדריכים את המשתמש בצורה שגויה בנוגע לפעילויות הדרושות בכל שלב, תוצאות ביניים בכל שלב, כולל סדר ורצף הפעילויות להשלמת התהליך.

לדוגמא: ממשק המשתמש בסיום עסקת רכישה מציג 3 אפשרויות משלוח ומפיק את מסמכי המשלוח לפי הבחירה: א. משלוח בדואר חבילות של הדואר, ב. משלוח בדואר שליחים של דואר ישראל, ג. משלוח באמצעות שליח שיגיע ללקוח תוך 24 שעות. עקב שגיאה בממשק המשתמש, בחירה באפשרות השנייה מפנה את החבילה באופן שגוי לשירות דואר החבילות של הדואר, ומדפיסה את מסמכי המשלוח בהתאם.

דוגמא נוספת: רשת למכירת חומרי בניין החליטה לתת הנחה של 5% ללקוחות שהרכישות החדשיות שלהם נטו (בניכוי החזרות) עולות על 100,000 ₪, ובתנאי שהחזרות באותו חודש לא עולות על 10% מהרכישות. חישוב הזכאות נעשה, באופן שגוי, בשני השלבים הבאים: בשלב הראשון אותרו הלקוחות שרכישותיהם החדשיות (A) עלו על 100,000 ₪. בשלב השני חושבו ללקוחות שאותרו בשלב הראשון, היקף ההחזרות באותו חודש (B). לקוחות ששיעור ההחזרות שלהם (B/A) עלה על 10% נופו מהרשימה של הזכאים להנחה. כך עקב תהליך שגוי, זכו בהנחה לקוחות שרכישותיהם נטו (A-B) קטנות מ-100,000 ₪.

ט. שגיאות תיעוד

שגיאות תיעוד הפוגעות בצוותי הפיתוח והאחזקה מצויות במסמכי התכנון, במדריכי תוכנה ובתיעוד המצוי בתוך התוכנה עצמה. הסתמכות על תיעוד שגוי עלול להביא לשגיאות נוספות בשלב הפיתוח והתפעול שוטף של התוכנה והאחזקה.

סוג אחר של שגיאות תיעוד משפיע בעיקר על המשתמשים ומקורו בשגיאות במדריכים למשתמש ובמידע "עזרה" (help) שנמצא ביישום התוכנה. שגיאות אופייניות מסוג זה הן:

- השמטת חלק מיישומי התוכנה.
- שגיאות בהסברים ובהנחיות הניתנות למשתמש המביאות את המשתמש למצב "ללא מוצא" (dead end) או לפעילות שגויה.
- מידע על תפקודי תוכנה (functions) שאינם קיימים, בדרך כלל כאלה שהוחלט בתהליך הפיתוח להפסיק את פיתוחם, או תפקודי תוכנה שהיו קיימים בגרסאות הקודמות של מוצר התוכנה והוחלט לבטלם.

בנושא גורמי השגיאות עוסקים Johnson and Mellou (2020) וגם Silva and Vieira (2017).

פרק 8:

עלות איכות תוכנה

בפרק זה:

- ⊙ הפעם אושר התקציב – מיני אירוע כמבוא
- ⊙ המטרות של מדידת עלויות איכות תוכנה
- ⊙ המודל הקלאסי של עלויות איכות תוכנה
- ⊙ היקפן של עלויות איכות התוכנה – נתוני התעשייה
- ⊙ המודל המורחב של עלויות איכות תוכנה
- ⊙ יישום של מערכת עלויות איכות תוכנה
- ⊙ קשיים ביישום מדידות של עלויות איכות תוכנה

8.1 הפעם אושר התקציב – מיני אירוע כמבוא

חנה שמואלי, ראש צוות הא"ת הייתה מאוד מאוכזבת כאשר המנכ"ל דחה בפעם השלישית את התוכנית להתקנת חבילת התוכנה Java Solver במחלקת פיתוח התוכנה.

חבילת התוכנה המוצעת הייתה בעלת יכולות מוכחות לזהות באופן אוטומטי 92 אחוזים של שגיאות תוכנת Java, ולתיקון אוטומטי של 85 אחוזים של השגיאות שזוהו. האיכות שנמצאה בניסויים של התוכנה המוצעת הייתה מדהימה. איכות התיקונים האוטומטיים הייתה דומה לאיכות התיקונים הידניים, ובחלק ניכר של התיקונים אפילו באיכות טובה מהאיכות של תיקונים ידניים.

חנה שמואלי החליטה להגיש בקשה נוספת (רביעית) לאישור התקנה של חבילת התוכנה במחלקת פיתוח התוכנה. המסמך התחיל בהצגת חבילת התוכנה ויכולותיה והצגת עלות השימוש ב-4 תחנות עבודה – 240,000 ₪ לשנה. חלק זה של המסמך היה גרסה משופרת של שלושת מסמכי הבקשה הקודמים. הכישלונות של הבקשות הקודמות הביא את חנה להוסיף הפעם לבקשה הוכחת כדאיות ההתקנה של חבילת התוכנה.

"מחקר" קצר של חנה שמואלי העלה את הממצאים הבאים:

1. מחלקת פיתוח התוכנה מעסיקה 10 מתכנני Java ו-10 בוחנים. עלות של מועסק היא 35,000 ₪ לחודש בממוצע.
2. נתונים סטטיסטיים של המחלקה הראו ש-6 מבין המתכנתים מייצרים תוכנה חדשה, ואילו 4 האחרים עסוקים בתיקונים לפי הממצאים של מבחני התוכנה.
3. על פי תוצאות הניסויים, מבחני התוכנה האוטומטיים שיידרשו למחלקה יוכלו להיעשות על ידי 4 בוחנים שיצוידו ב-4 תחנות עבודה של חבילת התוכנה.
4. עלות השימוש ב-4 תחנות עבודה – 240,000 ₪ לשנה.
5. עלות של קורס בן שבועיים להכשרת מתכנת בשימוש בתוכנת Java Solver היא 10,000 ₪ לכל עובד. עלות שכרו של העובד בתקופת הקורס – 17,500 ₪. מוצע להכשיר 6 עובדים בהפעלת חבילת התוכנה, כדי להבטיח זמינות עובדים בכל זמן.
6. לפי ההערכה יידרשו 3 בוחנים להשלים את מבחני התוכנה לאחר תהליך הבחינה האוטומטי של Java Solver. 2 מתכנתים שיתקנו את התוכנה לפי הממצאים השונים.
7. לסיכום, עם הפעלת חבילת התוכנה Java Solver, צוות המחלקה יהיה בן 8 מתכנתים ו-7 בוחנים, במקום 10 מתכנתים ו-10 בוחנים.

לאור נתונים אלה סיכום העלויות הוא כדלקמן:

עלות השימוש ב-4 תחנות העבודה 240,000 ₪ לשנה
עלות ההכשרה בשימוש בתוכנת Java Solver ל-6 עובדים . . 60,000 ₪.
עלות שכרם של 6 העובדים בתקופת ההכשרה 105,000 ₪.
החיסכון בהקטנת צוות המחלקה ב-5 עובדים 2,100,000 ₪ לשנה
החיסכון בשנה הראשונה להפעלת Java Solver 1,695,000 ₪ לשנה
החיסכון בשנה השנייה להפעלה ובשנים לאחר מכן 1,860,000 ₪ לשנה

חנה שמואלי הגישה את הגרסה הרביעית של בקשתה לאישור Java Solver שכללה הצגה של התחשיבים הכלכליים שהופיעו למעלה. ימים אחדים לאחר מכן הוזמנה חנה בדחיפות למשרדו של המנכ"ל.

המנכ"ל פתח את הפגישה "אנחנו, סמנכ"ל הכספים ואני, קראנו את ההצעה ביסודיות". הוא הפסיק לרגע, והמתיחות של חנה התגברה. "מצאנו את ההצעה שלך הצעת השקעה מצוינת, ומאשרים אותה לביצוע מיידית". המנכ"ל לא השאיר זמן לתגובתה של חנה ונעל את הישיבה כשהוא אומר "ההצעה הנוכחית שלך עוסקת בהוצאות הערכה. אשמח שההצעה הבאה שלך תעסוק בהוצאות מניעה".

יותר ויותר הנהלות – של חברות מסחריות או ארגונים ציבוריים – דורשים הערכות כלכליות של תוכניות הבטחת איכות המוגשות לאישורן. לפיכך, נעשה יותר ויותר רגיל שהצעות לפיתוח של כלי הא"ת חדשים, או השקעות לשיפור ולהרחבת מערכות פעילות, ייבחנו דרך "משקפיים כלכליות".

כך נעשה כחובה ליחידת הא"ת להציג את הרווחיות הפוטנציאלית של כל בקשה כספית משמעותית הדרושה עבור תשתית המערכת או להוצאות התפעול השוטף.

8.2 המטרות של מדידת עלויות איכות תוכנה

המטרה העיקרית של מדידת עלויות איכות תוכנה (costs of software quality) היא מתן אפשרות להנהלה להפעיל בקרה כלכלית על הפעילויות של הא"ת:

- בקרה של ההשקעות במניעה וזיהוי שגיאות.
- הערכת הנזקים של כשלי תוכנה.
- הערכת תוכניות להגדלה או צמצום של פעילויות הא"ת או להשקעות בעדכון או בחידוש תשתית הא"ת.

הפעלת הבקרה הניהולית באמצעות מדדי עלויות איכות תוכנה מבוססות בעיקר על השוואות עם:

- עלויות איכות תוכנה של השנה הקודמת.
- עלויות איכות תוכנה של פרויקט קודם.
- עלויות איכות תוכנה של מחלקות פיתוח תוכנה אחרות.

היחסים הכלכליים הבאים הם מצביעים חשובים להצלחת תוכניות הא"ת:

- אחוז עלויות איכות התוכנה מכלל עלויות פיתוח התוכנה.
- אחוז עלויות כשל איכות מתוך כלל עלויות פיתוח התוכנה.
- אחוז עלויות איכות התוכנה מתוך כלל הכנסות של מכירת מוצרי התוכנה ושירותי התוכנה.

לשם דוגמא, המצביע הראשון, במקרים רבים, מגיע לרמה של 50% מכלל עלויות הפיתוח של מוצר תוכנה. הפעלת כלי הא"ת שיוכלו להפחית את השיעור מ-50% ל-30%, תתרום באופן מובהק לשיפור הביצועים הכלכליים של בית התוכנה.

עלויות איכות תוכנה יכולות להיחשב כקבוצה מיוחדת של מדדי איכות תוכנה, כאשר ערכים כלכליים משמשים ככלי המדידה. בעוד שמדדי איכות תוכנה ועלויות איכות תוכנה מסייעים לבקרה הניהולית של הצדדים השונים של הא"ת, רק עלויות איכות תוכנה יכולים לספק מבט כולל על התוצאות של פעילויות הבטחת איכות התוכנה.

הנושא של עלויות פיתוח תוכנה הוא נושא למחקרים רבים, ספרים ומאמרים במהלך שני העשורים האחרונים. לעומת זאת מאמרים העוסקים בעלויות איכות תוכנה הם מעטים יחסית. נזכיר כאן שלוש דוגמאות: Karg (2009), Karg et al (2011) וכן Galin D. (2004). החשיבות של נתוני עלות איכות לשיפור תהליכים ארגוניים נדון בסעיף 4.7 של סטנדרט IEEE Std. 730-2014.

סעיף 9.3 דן במודל הקלאסי לעלויות איכות תוכנה, המיישם לתחום התוכנה מודל כללי לעלויות איכות. מודל נוסף, המודל המורחב לעלויות איכות תוכנה, המוצג בסעיף 9.5, הוא חלופה למודל הקלאסי המותאמת לתכונות הייחודיות של תעשיית התוכנה. סעיף 9.4 מציג נתונים על היקפי העלויות

של איכות התוכנה. החלק המסכם את הפרק מציג את שיטת היישום של מערכת עלויות איכות תוכנה (סעיף 9.6) ואת הקשיים העולים ביישום של מדידות עלויות איכות תוכנה (סעיף 9.7).

8.3 המודל הקלאסי של עלויות איכות תוכנה

המודל הקלאסי של עלויות איכות שפותח בתחילת שנות החמישים של המאה הקודמת על ידי Feigenbaum ואחרים, מספק מתודולוגיה כללית למיון עלויות הקשורות באיכות המוצר מנקודת ראות כלכלית. המודל פותח להתאים למאפיינים של ארגונים יצרניים, ומאז יושם באופן נרחב. המודל אומץ בתעשיית התוכנה, ונעשה למודל של עלויות איכות תוכנה.

המודל ממין את עלויות איכות התוכנה לשתי קבוצות כלליות: עלויות בקרה ועלויות כשל הבקרה.

עלויות בקרה (costs of control) – מתייחסות לעלויות המבוקרות על ידי מפתח התוכנה, וכוללות שתי תת קבוצות:

- **עלויות מניעה (costs of prevention)** – כוללות השקעות בתשתית ובפעילויות איכות המכוונות להפחתת שגיאות תוכנה. פעילויות אלה אינן מיועדות לפרויקט תוכנה מסוים, אלא לפעילויות שהן כלליות בארגון.

- **עלויות הערכה (costs of appraisal)** – הכוללות עלויות שמבוצעות בפרויקט תוכנה מסוים או מערכת תוכנה מסוימת למטרה של גילוי שגיאות תוכנה שיש צורך לתקן.

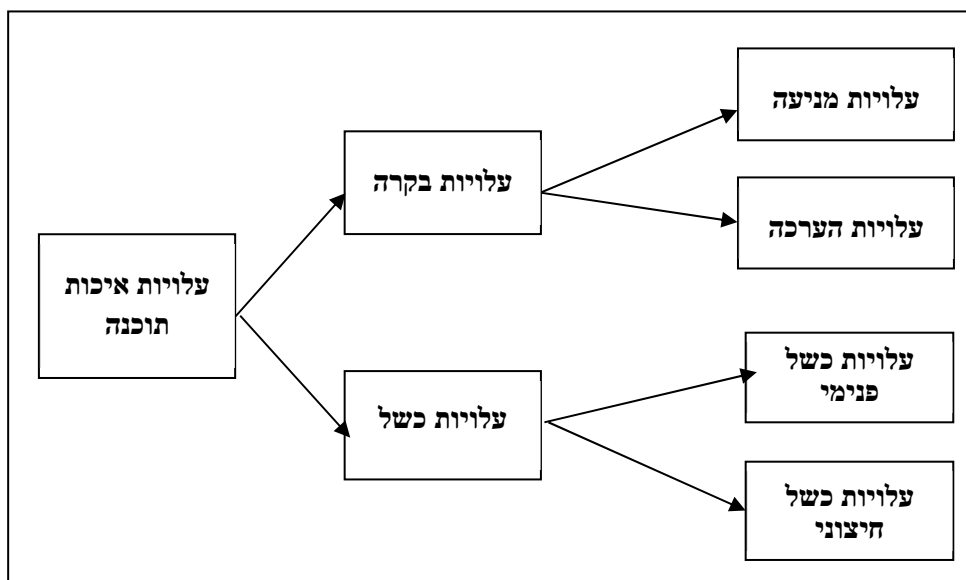
עלויות כשל הבקרה (costs of failure of control) – מתייחסות לעלויות של תיקון שגיאות תוכנה שהתגלו כתוצאה מפעילות הערכה או על ידי הלקוח. המודל מתפצל לשתי תת קבוצות:

- **עלויות כשל פנימי (costs of internal failure)** – הכוללות עלויות של תיקון שגיאות תוכנה שהתגלו על ידי סקרי תיכון, מבחני תוכנה, מבחני מערכת ומבחני קבלה, שתוקנו לפני שהמערכת הותקנה אצל הלקוח.

- **עלויות כשל חיצוני (costs of external failure)** – כוללות את כל העלויות של תיקון כשלי תוכנה המתגלים על ידי לקוחות או צוות האחזקה, לאחר שמערכת התוכנה הותקנה אצל הלקוח.

המודל הקלאסי של עלויות איכות תוכנה מוצג באיור 8.1.

איור 8.1: המודל הקלאסי של עלויות איכות תוכנה



בסעיפים הבאים נסקור את תת-הקבוצות של עלויות איכות התוכנה לפי המודל הקלאסי.

8.3.1 עלויות מניעה

עלויות מניעה כוללות השקעות בהקמה, עדכון ושיפור של תשתיות הא"ת, וכן עלויות של הפעולות הסדירות הדרושות לתפעול של התשתיות. חלק ניכר מפעילות יחידת הא"ת הן של מניעתיות, כפי שמופיע בתקציב הא"ת.

עלויות מניעה אופייניות כוללות:

(א) השקעות בפיתוח מרכיבי תשתית הא"ת או בשיפורה או בעדכון שוטף של מרכיבים אלה:

- נהלים והוראות עבודה.
- אמצעי עזר כגון תבניות ורשימות תיוג.
- מערכת בקרת תצורה.
- מדדי איכות תוכנה.
- מדידות של עלויות איכות תוכנה.

(ב) יישום סדיר של פעילויות הא"ת למניעה:

- הדרכת עובדים חדשים בנושאי הא"ת ונהלים הנוגעים לתפקידם.
- הדרכת עובדים בנושאי הא"ת ונהלים חדשים ומעודכנים.
- הסמכת עובדים לתפקידים מסוימים.
- ניתוח שגיאות ונתונים נוספים והביצוע בעקבותיהם של פעולות מונעות ומתקנות.

- ייעוץ על נושאי הא"ת שניתנו לראשי צוות ואחרים.
- (ג) בקרה של מערכת הא"ת על ידי ביצוע הפעולות הבאות:
 - מבדקי איכות פנימיים.
 - מבדקי איכות חיצוניים על ידי לקוחות וארגוני הסמכת מערכת הא"ת.
 - סקרי איכות של ההנהלה ("סקר הנהלה").

8.3.2 עלויות הערכה

עלויות הערכה מוקדשות לגילוי של שגיאות תוכנה בפרויקטים או במערכות תוכנה מסוימים. עלויות הערכה אופייניות כוללות:

- (א) עלויות סקרים:
 - סקר תיכון (DR).
 - סקרים המבוצעים על ידי עמיתים: ביקורת (inspection) וסריקה (walkthrough).
 - בדיקות מומחה.
- (ב) עלויות מבחני תוכנה:
 - מבחני יחידה (unit test).
 - מבחני אינטגרציה.
 - מבחני מערכת.
 - מבחני קבלה (עלויות ההשתתפות במבחנים המבוצעים על ידי הלקוח).
- (ג) עלויות הבטחת איכות של עבודה המבצעת על ידי גורמי חוץ, בעיקר על ידי ביצוע סקרי תיכון ומבחני תוכנה ובקרת התקדמות פרויקטים. גורמי חוץ כוללים:
 - קבלני משנה.
 - ספקי חבילות תוכנה ומודולי תוכנה ממוחזרים.
 - הלקוח (כאשר הוא משתתף בביצוע הפרויקט).

8.3.3 עלויות כשל פנימי

עלויות כשל פנימי הן עלויות לביצוע תיקון שגיאות תוכנה שנתגלו בסקרי תיכון, מבחני תוכנה ומבחני קבלה שנעשו לפני התקנת התוכנה אצל הלקוח. כאן המקום להעיר כי תיקונים ושינויים הנעשים על ידי אנשי הצוות בעקבות בדיקת ראש הצוות או הערות עמיתים לצוות, אינן עלויות כשל פנימי אלא חלק אינטגרלי מתהליכי התיכון והתכנות ומהוות חלק מעלויות פיתוח תוכנה.

סוגי עלויות אופייניים של כשל פנימי הן:

- עלויות תכנון מחדש ותכנות מחדש (rework) כתוצאה מממצאי סקרי תיכון ומבחני תוכנה.
- עלות תיקוני תיכון ותכנות כתוצאה מממצאי סקרי תיכון ומבחני תוכנה.

- עלות סקרי תיכון חוזרים ומבחני תוכנה חוזרים (מבחני רגרסיה, regression tests). כדאי לציין כי בעוד עלויות של סקרי תיכון ומבחני תוכנה הם עלויות הערכה, הרי שעלות סקרי תיכון חוזרים ומבחני תוכנה חוזרים היא עלות כשל פנימי.
- נזקים של "אפקט הדומינו". איחורים בהשלמת פרויקט תוכנה גורמים נזקים לפרויקט המאחר, אבל בנוסף, גורמים נזקים לפרויקטים נוספים הממתינים לאנשי צוות שהיו אמורים להשתחרר מהפרויקט המאחר. הנזקים עלולים להתגלגל לפרויקטים מאוחרים יותר התלויים באנשי צוות מאחד מהפרויקטים הקודמים שהתאחרו. במילים אחרות, כאשר פרויקט מתאחר אנחנו יכולים לצפות ל"אפקט הדומינו" כאשר שורה של פרויקטים כל אחד יתאחר באשמת הפרויקט הקודם לו.

8.3.4 עלויות כשל חיצוני

עלויות כשל חיצוני נגרמות על ידי תקלות תוכנה שהתגלו על ידי לקוחות או צוותי אחזקה לאחר שמערכת התוכנה הותקנה אצל הלקוח. עלויות אלה נחלקות בין עלויות "גלויות" של כשל חיצוני לבין עלויות "חבויות" של כשל חיצוני. עלויות הכשל החיצוניות ה"גלויות" מתייחסות לעלויות פיזיות של תיקון תקלות תוכנה שהתגלו ולתשלומי פיצוי שנעשו עקב תקלות תוכנה או איכות נמוכה של מוצר התוכנה. עלויות הכשל החיצוניות ה"חבויות" מתייחסות לנזקים שנגרמו למוניטין ולכוחר התחרותיות של המפתח עקב תקלות תוכנה חמורות, ובעיקר עקב פרסומן.

עלויות "גלויות" אופייניות של כשל חיצוני הן:

- עלויות פתרון של תלונות לקוחות במהלך תקופת האחריות. פתרון תלונות כאלה מתקבלות באמצעות הטלפון או בתקשורת באמצעות האינטרנט או אחרת. הטיפול בתלונה כזאת דורשת לימוד פרטי התלונה ומתן הוראות לפתרון הבעיה. ברוב המקרים התלונה נובעת מכשל של הפנייה ל"עזרה" (help) שבתוכנה או של הוראות המדריך למשתמש.
- עלויות תיקון של תקלות תוכנה המתגלות במהלך התפעול השוטף. בתיקון התקלות נכללים מבחני התוכנה של התיקונים. חלק מהתיקונים מבוצעים באתר הלקוח.
- עלויות תיקון כשלי תוכנה לאחר תקופת האחריות, כאשר התיקון איננו נכלל באחריות.
- עלות פיצויים המשולמים ללקוח עקב כשלי תוכנה חמורים המתגלים בתפעול השוטף.
- עלות פיצויים המשולמים ללקוח עקב איחור בהשלמת הפרויקט.
- עלות החזר מחיר התוכנה ללקוח במקרים של אי שביעות רצון מוחלטת של הלקוח ממוצר התוכנה.
- עלות ביטוח כנגד תביעות לקוח. בפרויקטים שהמפתח מבוטח כנגד תביעות לקוח, הביטוח מכסה חלק מסוגי התביעות שנזכרו לעיל.
- עלויות כשל חיצוני לפי "אפקט הדומינו", שברוב המקרים מתייחסות לקנסות איחור שיוטלו על פרויקטים שנפגעו מ"אפקט הדומינו".

עלויות כשל תוכנה חיצוני שאינן מדווחות

במקרים לא מעטים למפתח יש עניין ברור "להפחית" את הוצאות כשל התוכנה החיצוני. ניתן להשיג זאת על ידי החלפת תשלומי פיצוי על נזק ללקוח ב"הסדר" שלא יירשם כעלות כשל חיצוני. דרכים אפשריות לעשות זאת הן:

- ויתור על התשלום האחרון של הלקוח.
- ויתור על חיוב הלקוח על ביצוע שינויי תוכנה שנדרשו על ידי הלקוח.
- ויתור על חיוב הלקוח על שירותי אחזקת מערכת התוכנה לאחר תום תקופת האחריות.
- מתן הנחות ללקוח על הזמנות עתידיות למוצרי תוכנה.

הערך הכספי של "הסדרים" כאלה לא מופיע בהנהלת החשבונות של החברה. וכך לא נכלל בעלויות הכשל החיצוניות.

דוגמאות אופייניות של עלויות כשל חיצוני "חבויות" הן:

- נזקים מהקטנת מכירות ללקוחות שסבלו מתקלות תוכנה.
- הקטנה במכירות כתוצאה של פגיעה במוניטין של החברה כתוצאה מתקלות תוכנה חמורות שהתפרסמו.
- הוצאות מוגדלות לקידום מכירות, כדי להקטין את ההשפעות השליליות של תקלות התוכנה החמורות שהתפרסמו.
- הקטנת הסיכויים לזכייה במכרזים עקב הפגיעה במוניטין.
- הצורך להנמיך את המחיר בהצעות כדי להתגבר על הירידה במוניטין ולהגדיל את הסיכוי לזכות במכרזים.

ברוב המקרים ההיקף של עלויות "חבויות" גדול בהרבה מהעלויות ה"גלויות". מאפיין של עלויות הכשל החיצוני ה"חבויות" הוא הקושי לאמוד את היקפן בהשוואה לעלויות הכשל החיצוני ה"גלויות" הרשומות בהנהלת החשבונות והיקפן ידוע. בנוסף, פעמים רבות יש מחלוקות, בין אנשי המקצוע המעורבים, ביחס להיקפים של עלויות הכשל החיצוני ה"חבויות". כתוצאה, לעיתים קרובות, לא מבצעים אומדן של עלויות "הכשל החיצוני ה"חבויות".

8.4 היקפן של עלויות איכות התוכנה – נתוני התעשייה

הנושא של עלויות פיתוח תוכנה הוא נושא למחקרים רבים, לעומת זאת מאמרים העוסקים בעלויות איכות תוכנה, ובמיוחד כאלה המציגים נתונים שנאספו, הם מעטים ביותר.

מודלים של עלויות איכות תוכנה ושל חלקן בכלל עלויות מוצר התוכנה הם נושא של מאמרים אחדים, כולל Daughtrey T. (2013), Jones C. (2011), Krasner, Laporte et al (2012), Galin (1998), וגם Galin and Avrahami (2007) ו-Galin (2004).

מחקרים אחדים שהתפרסמו עוסקים בנושא הקשר שבין רמת איכות מערכת פיתוח התוכנה של הארגון לפי מתודולוגיית CMM והחלק היחסי של עלויות איכות התוכנה מכלל עלויות הפיתוח. Knox (1993) מצא שעלויות איכות התוכנה מהוות 60% מעלויות פיתוח התוכנה בארגונים שרמתם נמוכה (CMM רמה 1), ויורדות ל- 51% כשרמתם בינונית (CMM רמה 3) ומגיעות ל- 23% כשרמת הארגון היא הגבוהה ביותר (CMM רמה 5). התוצאות המפורטות שקיבל Knox, הכוללות חלוקת עלויות האיכות לפי ארבעת המרכיבים של מודל הקלאסי של עלויות איכות התוכנה מופיעות בטבלה 8.1.

טבלה 8.1: עלויות איכות התוכנה לפי רמת CMM

רמת CNN					עלויות
5	4	3	2	1	
23%	36%	51%	57%	60%	עלות איכות תוכנה כחלק מסך עלות פיתוח התוכנה
12%	7%	4%	2%	2%	עלות מניעה
4%	11%	12%	10%	4%	עלות הערכה
5%	15%	25%	25%	22%	עלות כשל פנימי
2%	3%	10%	20%	32%	עלות כשל חיצוני

נתונים עדכניים יותר על היקף עלויות האיכות תוכנה בפרויקטים של פיתוח תוכנה באירופה ואמריקה פורסמו על ידי Laporte et al (2012), ומוצגים בטבלה 8.2.

טבלה 8.2: התפלגות עלויות איכות התוכנה לפי Laporte et al

פרויקטים של פיתוח תוכנה	פרויקטים אמריקנים (2 פרויקטים)	פרויקטים אירופאיים (3 פרויקטים)
סך כל עלות איכות תוכנה (כחלק מעלות הפרויקט)	57.5%	67.7%
עלויות מניעה	13.5%	8.7%
עלויות הערכה	16%	26.3%
עלות כשל (פנימי וחיצוני)	28%	32.7%

8.5 המודל המורחב של עלויות איכות תוכנה

ניתוח עלויות איכות התוכנה כפי שהוגדרו במודל הקלאסי מגלה כי מספר סוגי עלויות בעלות גודל משמעותי לא נכללו במודל. עלויות אלה הן ייחודיות לתעשיית התוכנה, אך בעלות גודל זניח בתעשיות האחרות. לדוגמא, סוגי עלויות כשל הבאות:

- נזקים ששולמו ללקוח על איחור בהשלמת פרויקט תוכנה, שנגרמו כתוצאה מלוח זמנים לא ריאלי לפרויקט.
- נזקים ששולמו ללקוח על איחור בהשלמת פרויקט תוכנה, שנגרמו כתוצאה מכישלון בגיוס צוות.

המשותף לשתי הדוגמאות הוא שהעלויות שנגרמו, שבמקרים רבים הן גבוהות, אינן תוצאה של פעילות או חוסר מקצועיות של צוות הפיתוח אלא תוצאה של כשלון ההנהלה. מאפיין משותף נוסף הוא היקף העלויות, בעוד שבתעשיות היצרניות, העוסקות ביצור סדרתי של מוצרים, העלויות הן זניחות, בתעשיית התוכנה מקרי הכשל הניהולי האלה שכיחים למדי והפיצויים המשולמים הם משמעותיים.

עיון נוסף בעלויות איכות מעלה דוגמאות נוספות לפעילויות של ההנהלה למניעה או הפחתה של עלויות איכות:

- ביצוע סקר חוזה תוכנה. ביצוע סקר טיוטת ההצעה וסקר טיוטת החוזה עשויים למנוע מצבים של תמחור ואומדן לוח זמנים לא נכונים, דבר שיביא, בנוסף להפסד כספי, לביצוע תחת לחץ תקציבי ולחץ לוח זמנים. מצבי לחץ כאלה הם מתכון לכשל איכות בפרויקט. גם כאן, ביצוע סקר חוזה הוא פעולה של ההנהלה ואינן כלל בתחום פעילות צוות הפרויקט.
- הכנת תוכנית פרויקט תוכנה ותוכנית איכות צריכות להיעשות על ידי מנהל שהפרויקט לפני התחלת ביצוע הפרויקט. גם כאן הפעולה של הכנת התוכנית היא באחריות ההנהלה ולא מתפקידו של צוות הפרויקט.
- הפעלה נכונה של בקרת התקדמות פרויקט תוכנה. בקרת התקדמות המופעלת נכון עשויה לאתר בשלב מוקדם קשיים בביצוע הפרויקט ולפתור אותם תוך הקטנת הנזק או מניעתו. מטבע הדברים ביצוע בקרת ההתקדמות היא פעולת הנהלה שאיננה באחריות צוות הפרויקט.

המשותף לשלוש הדוגמאות הוא שגם כאן הפעולה היא באחריות ההנהלה ולא מתפקידו של צוות הפרויקט. מאפיין משותף נוסף הוא היקף העלויות, בעוד שבתעשיות היצרניות, העוסקות בייצור סדרתי של מוצרים, העלויות הן זניחות, בתעשיית התוכנה עלות כוח האדם המקצועי לביצוע הפעולות היא משמעותית ביותר.

אחריות ההנהלה על עלויות איכות תוכנה מוצגת בספר של Flowers כבר בכותרתו "כשלון תוכנה: כשלון ההנהלה" (*Software Failure: Management Failure*), (Flowers 1996).

המודל המורחב לעלויות איכות תוכנה

המודל המורחב לעלויות איכות תוכנה, שפותח על ידי המחבר של ספר זה, נותן ביטוי לעלויות איכות תוכנה שהנהלה "תורמת" לכלל העלויות. המודל המורחב מוסיף לעלויות איכות התוכנה הקשורות בצוות הפיתוח, והמופיעות במודל הקלאסי של עלויות איכות התוכנה, את העלויות הקשורות בפעולות ההנהלה בקשר לביצוע הפרויקט. לפי המודל המורחב, כלל העלויות מסווג לשתי קבוצות:

א. **עלויות איכות של פיתוח התוכנה ופעילויות הא"ת**, המוגדרות על פי המודל קלאסי של עלויות איכות תוכנה.

ב. **עלויות ניהוליות של איכות תוכנה** (management costs of software quality).

המודל ממייך את עלויות איכות התוכנה הניהוליות לשתי קבוצות כלליות: עלויות בקרת תוכנה ניהולית ("עלויות בקרה ניהולית") ועלויות כשל הבקרה הניהולית ("עלויות כשל ניהולי").

עלויות בקרה ניהולית (management control costs) – מתייחסות לעלויות הבקרה של פיתוח התוכנה המבוצעות על ידי ההנהלה, וכוללת שתי תת-קבוצות:

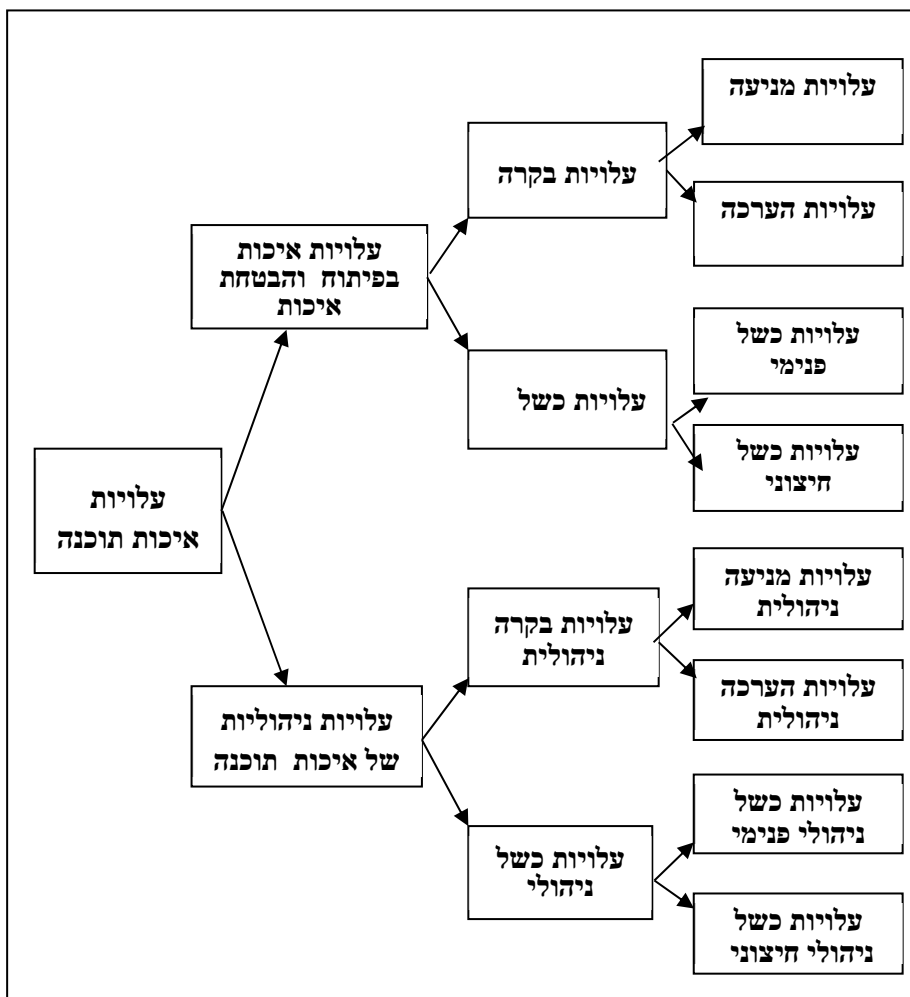
- **עלויות מניעה ניהולית** (management prevention costs) הכוללות השקעות בפעולות הבטחת איכות הפרויקט המבוצעות על ידי ההנהלה לפני התחלת ביצוע הפרויקט,
- **עלויות הערכה ניהולית** (management appraisal costs) הכוללות עלויות בקרת התקדמות הפרויקט מסוים.

עלויות כשל ניהולי (management failure costs) – מתייחסות לעלויות הנגרמות מכשל בעלויות המניעה וההערכה הניהוליות ובפעילויות המוטלות על ההנהלה כגון כישלון גיוס עובדים לפרויקט. המודל מתפצל לשתי תת קבוצות:

- **עלויות כשל ניהולי פנימי** (internal management failure costs) הכוללות עלויות עודפות של פיתוח התוכנה שנגרמו כשל בפעולות המניעה הניהוליות ובעלויות ההערכה הניהוליות.
- **עלויות כשל ניהולי חיצוני** (external management failure costs) כוללות את כל העלויות של פיצויי הלקוחות כתוצאה מכשל בפעילויות שבאחריות ההנהלה.

המודל המורחב של עלויות איכות תוכנה מוצג באיור 8.2.

איור 8.2: המודל המורחב של עלויות איכות תוכנה



בסעיפים הבאים מוצגות תת הקבוצות של העלויות הניהוליות של איכות התוכנה.

8.5.1 עלויות מניעה ניהולית

עלויות מניעה ניהולית עוסקות בפעילויות המבוצעות למנוע כשל ניהולי או להקטין את הסיכוי שהכשל הניהולי יקרה. פעילויות אלה צריכות להתבצע לפני תחילת העבודה בפרויקט תוכנה, והן באחריות ההנהלה. רוב הפעילויות האלה נדונות בהרחבה בפרקים אחרים של הספר.

עלויות אופייניות של מניעה ניהולית כוללות:

- עלויות ביצוע סקרי חוזה.
- עלויות הכנת תוכנית פרויקט תוכנה ותוכנות איכות, וביצוע סקרי תיכון לבדיקתן.
- עלויות ביצוע סקרי חוזה להצעות של קבלני משנה המיועדים לבצע חלקים מפרויקט התוכנה.

8.5.2 עלויות הערכה ניהולית

עלויות הערכה ניהולית קשורות בבקרה הניהולית של ביצוע פרויקטים של תוכנה, מיועדות לאתר במועד מוקדם סטיות ואיחורים בתוכנית העבודה. איתור מוקדם מאפשר תיקון החריגה בעלות נמוכה או אף ללא עלות. נושאי בקרת התקדמות הפרויקט נדון בהרחבה בפרק אחר.

עלויות אופייניות של הערכה ניהולית כוללות:

- עלויות ביצוע עדכון תקופתי של תוכנית הפרויקט ותוכנית האיכות.
- עלות ביצוע בקרה תקופתית של התקדמות עבודת צוות הפרויקט.
- עלות ביצוע בקרה תקופתית של התקדמות עבודת גורמי חוץ המעורבים בביצוע הפרויקט.

8.5.3 עלות כשל ניהולי פנימי

עלויות כשל ניהולי פנימי עלולות להיגרם בכל מהלך ביצוע הפרויקט. הן עלולות להיגרם עקב כישלון בזיהוי סיכונים לפרויקט, כישלון בהכנת תוכנית הפרויקט, אשר כולם כישלונות ההנהלה.

עלויות אופייניות של כשל ניהולי פנימי כוללות:

- עלויות נוספות בלתי מתוכננות למשאבי אנוש ואחרים, הנובעות מהערכות חסר שעליהן התבססו הצעות פרויקט.
- עלויות מוגדלות ששולמו לקבלני משנה עבור עובדים שגויסו בתנאי לחץ שנוצרו בתוקף הנסיבות.
- עלויות צוות נוספות של "גיוס פנימי" שנעשה ברגע האחרון, כאשר הצוות שגויס לא היה המתאים ביותר לתפקיד.
- נזקי "אפקט הדומינו" פנימיים לפרויקט לפני מסירת מוצר התוכנה ללקוח, וזאת עקב כשל ניהולי בפרויקט אחר.

8.5.4 עלויות כשל ניהולי חיצוני

כצפוי, רוב עלויות הכשל הניהולי החיצוני נגרמות לאחר השלמת פיתוח התוכנה והתקנתה אצל הלקוח. עלויות אלה נגרמות עקב איחור בהשלמת הפרויקט או כשל באיכות הפרויקט, כאשר הגורם הוא פעילות לא נאותה של ההנהלה או היעדר פעילות שנדרשה.

עלויות אופייניות של כשל ניהולי חיצוני כוללות:

- פיצויים ששולמו ללקוח על איחורים בהשלמת הפרויקט, שנגרמו עקב לוח זמנים לא ריאלי בהצעת הפרויקט.
- פיצויים ששולמו ללקוח על איחורים בהשלמת הפרויקט, שנגרמו עקב כשלון ההנהלה בגיוס אנשי צוות מתאימים.
- נזקי "אפקט הדומינו". תשלום פיצויים ללקוח על איחור בהשלמת הפרויקט בגלל "אפקט הדומינו" שנוצר בפרויקט אחר שחל בו כשל ניהולי.

8.6 יישום של מערכת עלויות איכות תוכנה

כדי ליישם מערכת עלויות איכות בארגון יש לבצע:

- הגדרת מודל של עלויות איכות תוכנה ופריטי עלות מתאימים.
- הגדרת שיטות איסוף נתונים.
- הפעלת המערכת, דהיינו איסוף הנתונים לפי המודל שנקבע, כולל בקרת הביצוע.
- נקיטת פעולות בתגובה לממצאים.

8.6.1 הגדרת מודל של עלויות איכות תוכנה ופריטי עלות מתאימים

בשלב ראשון על הארגון לבחור במודל עלות איכות תוכנה המתאים לו: הקלאסי, המורחב או אחר. לאחר מכן יש לקבוע פריטי עלות שיימדדו עבור המודל. פריטי עלות אלה יהיו ייחודיים לארגון ויבטאו את סוגי הפעילויות והוצאות התקציב הנהוגים בארגון. כל פריט עלות חייב להיות משויך לאחד מתת הקבוצות המופיעות במודל

דוגמא

יחידת הא"ת בבית תוכנה גדול החליטה להקים מערכת לעלות איכות תוכנה, ובחרה במודל הקלאסי כמודל העלות איכות תוכנה של החברה. איש צוות היחידה הכין רשימה של 13 פריטי עלות שמוינו על ידי ראש היחידה לפי תת הקבוצות שבמודל. 3 מבין פריטי העלות נדחו על ידי ראש היחידה שקבע שהם פריטי עלות של פיתוח התוכנה ולא של עלות איכות התוכנה. תוצאות המיון של פריטי העלות מוצגים בטבלה 8.3.

טבלה 8.3: מיון של פריטי עלות איכות תוכנה פוטנציאליים – דוגמא

פריט העלות	תת קבוצה במודל המורחב
1. עלות ההעסקה של ראש יחידת הא"ת	50% עלות מניעה 50% עלות הערכה
2. עלות האימות של מילוי הוראות נוהלי הא"ת על ידי חברי צוות הפרויקט	עלות מניעה
3. תשלום לקבלן משנה עבור הכנת היעוד הפרויקט	לא עלות איכות תוכנה (עלות פיתוח תוכנה)
4. עלות השתתפות במבדקי איכות פנימיים וחיצוניים בצוותי הפיתוח והאחזקה	עלות מניעה
5. עלות פיצוי ללקוח על איחור במסירת פרויקט התוכנה כתוצאה מאיחור של 7 שבועות בהתחלת ביצוע הפרויקט.	עלות כשל ניהולי חיצוני
6. עלות מבחני התוכנה הראשונים שבוצעו על ידי יחידת הבחינה	עלות הערכה
7. עלות מבחנים חוזרים (מבחני רגרסיה) שבוצעו על ידי יחידת הבחינה	עלות כשל פנימי
8. עלות תיקון שגיאות שהתגלו ע"י צוות הבחינה	עלות כשל פנימי
9. עלות תיקונים על ידי צוות האחזקה של כשלי תוכנה שנתגלו על ידי הלקוח	עלות כשל חיצוני
10. עלות שעות העבודה שהשקיע מנהל מחלקת התוכנה בישיבות מעקב התקדמות הפרויקט	עלות הערכה ניהולית
11. עלות בדיקות ראש הצוות של תוכנה שפותחה על ידי אנשי הצוות.	לא עלות איכות תוכנה (עלות פיתוח התוכנה)
12. עלות היועץ של יחידת הא"ת	עלות מניעה
13. עלות יועץ שהשתתף בסקר חוזה הפרויקט	עלות מניעה ניהולית
14. עלות השתתפות היועץ של יחידת הא"ת בחקירת כשלים חיצוניים	עלות כשל חיצוני
15. עלות רכישת תוכנת אבטחת נתונים עבור יחידת שירות לקוחות	לא עלות איכות תוכנה (עלות פיתוח התוכנה)
16. עלות רכישת ירחונים של הא"ת והשתתפות בכנסים של הא"ת	עלות מניעה

הצעה ליישום
מחלקות הפיתוח והאחזקה חייבות להגיע להסכם על מודל עלות איכות התוכנה ועל פריטי העלות. מומלץ לוותר על פריטי עלות שאינם מוסכמים או שקשה לאסוף את הנתונים עבורם, ואפילו במחיר של כיסוי נמוך יותר של עלויות איכות התוכנה. ככל שמערכת עלות התוכנה מוסכמת ופשוטה יותר לאיסוף הנתונים, כך גדלים הסיכויים להפעלה מוצלחת של המערכת.