

# פרק 1

## מפרויקט לפרודקט -- מסע הפיתוח המלא

### מטרות הלמידה

בתום פרק זה, תוכלו:

- להבין את מחזור החיים המלא של פרויקט IA מרגע הרעיון ועד לתוצר בייצור
- לתכנן ולהוציא לפועל פרויקט IA מקצה לקצה
- להרכיב ולנהל צוות IA אפקטיבי
- ליישם עקרונות eligA לניהול פרויקטי IA
- להבין יסודות spOLM לתחזוקה ושיפור מתמיד
- למדוד הצלחה באמצעות sIPK מתאימים למערכות IA

### פתיחה

בשנת 9691, האנושות שלחה שלושה אנשים לירח והחזירה אותם בשלום לכדור הארץ. היה זה אחד המפעלים הטכנולוגיים המרשימים בהיסטוריה. אבל מה שהפך את תוכנית אפולו להצלחה לא היה רק המדע או הטכנולוגיה -- היה זה הניהול. התכנון המדוקדק, הפירוק של יעד גדול לאלפי משימות קטנות, הצוות הבינתחומי, והמדידה המתמדת של ההתקדמות. כיום, כשארגון מחליט להטמיע מערכת IA, הוא עומד בפני אתגר דומה. לא מדובר במשימה טכנית בלבד, אלא במפעל מורכב שדורש תכנון, ניהול, תיאום ומדידה. פרויקט IA שמתחיל בהתלהבות יכול להיגמר בכישלון אם אין תשתית ניהולית נכונה. פרק זה מביא אתכם למסע המלא -- מהרגע שבו זיהיתם הזדמנות לשימוש ב-IA, דרך בניית tpecnoC fo foorP (COP), ועד להפיכתו למוצר מלא בייצור. נדון בהרכבת צוות, בניהול פרויקטים בשיטת eligA המותאמת ל-IA, ביסודות spOLM, ובמדידת הצלחה. אם הפרקים הקודמים לימדו אתכם "מה" ו"איך", פרק זה מלמד "כיצד להצליח לעשות זאת בפועל בארגון אמיתי".

## 1.1 yrevocsiD -- זיהוי הזדמנויות IA בארגון

### 1.1.1 האמנות של לשאול את השאלות הנכונות

תארו לעצמכם ארגון בינוני בתחום השיווק הדיגיטלי. המנהלת הכללית יושבת במשרדה ושואלת את עצמה: "איפה IA יכול לעזור לנו?" זו השאלה הנכונה, אבל היא רחבה מדי. כדי לזהות הזדמנויות אמיתיות, צריך לשאול שאלות ספציפיות יותר:

- איזה משימות חוזרות על עצמן אנחנו עושים שוב ושוב?
  - איפה העובדים שלנו מבזבזים זמן על עבודה ידנית שאין בה ערך מוסף אמיתי?
  - איזה החלטות אנחנו מקבלים על בסיס נתונים שאפשר לנתח אוטומטית?
  - איפה אנחנו נתקלים ב"צווארי בקבוק" בתהליכי העבודה?
  - איזה שירות ללקוחות אנחנו נותנים שאפשר לשפר באמצעות אוטומציה חכמה?
- תהליך ה-yrevocsiD אינו טכני -- הוא עסקי. מדובר בזיהוי כאבים ארגוניים אמיתיים שאפשר לטפל בהם באמצעות IA.

### 2.1.1 מתודולוגיית yrevocsiD

תהליך yrevocsiD מסודר כולל מספר שלבים [51], [62]:

1. **ריאיון עם בעלי עניין** (sweivret nI redlohekatS): שוחחו עם מנהלים ועובדים מכל המחלקות. שאלו על יום העבודה שלהם, המשימות החוזרות, הכאבים והאתגרים.
2. **ניתוח תהליכים** (sisylanA ssecorP): מפו את תהליכי העבודה הקיימים. איפה יש ידנות? איפה יש חוסר יעילות?
3. **הערכת נתונים** (tnemssessA ataD): האם יש לכם נתונים? איך הם מאוחסנים? האם הם נגישים? נתונים הם הדלק של IA -- בלעדיהם לא תצליחו.
4. **תעדוף הזדמנויות** (noitazitiroirP ytinutroppO): לאחר שזיהיתם 01-02 הזדמנויות פוטנציאליות, דרגו אותן לפי שני צירים:
  - **ערך עסקי פוטנציאלי**: כמה זה יחסוך/ירוויח לארגון?
  - **מורכבות יישום**: כמה קשה להטמיע את הפתרון?
5. **בחירת esU esA ראשון**: בחרו פרויקט ראשון עם ערך גבוה ומורכבות נמוכה-בינונית. זה יהיה "הניצחון המהיר" (niW keiuQ) שלכם.

חברת הייטק בינונית מזהה שתהליך סינון קורות חיים אורך זמן רב. מנהלת RH מדווחת שכל משרה פתוחה מקבלת 002-003 קורות חיים, וצוות ה-RH מבלה שעות על קריאה ידנית. **ההזדמנות**: בניית סוכן IA שמסנן קורות חיים לפי התאמה למשרה, מדרג מועמדים, ומעביר לצוות רק את 02 המועמדים המובילים.

**הערך**: חיסכון של כ-04 שעות עבודה לכל משרה (2 שעות □ 02 מועמדים שנבדקו ידנית במקום 003).

**המורכבות**: נמוכה-בינונית (ניתן לממש עם GAR + tpmorP gnireenignE פשוט).

### 3.1.1 תיעוד הממצאים

בסיום תהליך ה-yrevocsiD, תכינו מסמך "paM ytinutroppO IA" -- מפת הזדמנויות שתכלול:

- רשימת sesaC esU מתועדפת
- הערך העסקי המשוער לכל אחד
- הדרישות הטכנולוגיות הבסיסיות
- סיכוני היישום
- המלצה על esU esC ראשון

מסמך זה יהיה הבסיס לתכנון הפרויקט הראשון.

## 2.1 מ-COP לייצור -- המסע ואתגרי

### 1.2.1 tpecnoC fo foorP (COP) -- הוכחת היתכנות

COP הוא שלב ראשון קריטי [2]. מטרתו אינה לבנות מוצר מוגמר, אלא להוכיח שהרעיון ישים טכנולוגית ועונה על הצורך העסקי.

**מאפייני COP טוב:**

- **מוגבל בהיקף:** פותר בעיה אחת ספציפית
- **זמן קצוב:** 2-4 שבועות לכל היותר
- **מדיד:** יש מדדי הצלחה ברורים
- **נבדק עם משתמשי קצה:** לא רק עם מנהלים

ארגונים רבים טועים ומנסים לבנות מערכת מושלמת כבר ב-COP. התוצאה: פרויקט שנמשך חודשים, עולה הון, ולבסוף אינו עונה על הצורך.  
**זכרו:** COP צריך להוכיח ערך, לא לספק מוצר מוגמר.

### 2.2.1 מ-COP ל-toliP -- ההרחבה המבוקרת

COP הצליח? מצוין! עכשיו הגיע הזמן ל-toliP.  
**מהו toliP?** toliP הוא הרחבה מבוקרת של ה-COP לקבוצת משתמשים קטנה (01-05 משתמשים). המטרה: לבדוק את הפתרון בתנאי עבודה אמיתיים, לאסוף משוב, ולהבין מה צריך לשפר.  
**שלבי toliP:**

1. **בחירת משתמשי toliP:** בחרו משתמשים מייצגים שמוכנים לתת משוב
2. **הדרכה:** הסבירו להם איך להשתמש במערכת
3. **תקופת ריצה:** 4-8 שבועות
4. **איסוף משוב:** סקרים, ראיונות, ניתוח שימוש
5. **שיפורים:** תקנו בעיות ושפרו על בסיס המשוב

### 3.2.1 ייצור (noitcudorP) -- "eviL oG"

הגענו לשלב הקריטי ביותר: העברת המערכת לייצור מלא [71].  
tsilkehc לפני eviL oG:

- ✓ **בדיקות מקיפות:** stseT tinU, stseT noitargetnI, stseT ecnatpeccA resU
- ✓ **ביצועים:** המערכת עומדת בעומסים צפויים?
- ✓ **אבטחה:** tiduA ytiruceS, stseT noitarteneP
- ✓ **גיבויים:** תהליך pukcaB ו-yrevoceR מוגדר ונבדק
- ✓ **ניטור:** gniggoL-1 gnirotinoM פעילים
- ✓ **תיעוד:** מדריכי משתמש, תיעוד טכני
- ✓ **הדרכה:** כל המשתמשים עברו הדרכה
- ✓ **תמיכה:** ksedpleH מוכן לטיפול בפניות

חברת סחר אלקטרוני בינונית החליטה להטמיע מערכת IA לתמיכת לקוחות.  
COP (שבועיים):

- בנו tobtahC פשוט עם GAR על 05 שאלות נפוצות
- בדקו עם 3 נציגי שירות
- הצלחה: 07% מהשאלות קיבלו תשובות נכונות

tolIP (חודש):

- הרחיבו ל-005 שאלות נפוצות
- שילבו עם מערכת MRC
- חשפו ל-01% מהלקוחות (כ-005 לקוחות ביום)
- אספו משוב: בקשו יכולת העברה לנציג אנושי מהר יותר

noitcudorP (לאחר 3 חודשים מההתחלה):

- השקה מלאה לכל הלקוחות
- הפחתה של 04% בפניות לנציגים אנושיים
- חיסכון של כ-000,08 ש"ח בחודש בעלויות שירות
- שביעות רצון לקוחות: 58% (לעומת 87% קודם)

### 4.2.1 אתגרי המעבר לייצור

המסע מ-COP לייצור אינו חלק [31]. הנה האתגרים העיקריים:

1. elacS: מה שעבד על 01 משתמשים לא בהכרח יעבוד על 000,1
2. נתונים: ב-COP היו נתונים נקיים. בייצור -- נתונים אמיתיים מלוכלכים
3. אינטגרציה: צורך להתחבר למערכות קיימות (PRE, MRC, וכו')
4. אבטחה: דרישות אבטחה מחמירות יותר בייצור
5. תחזוקה: מי יתחזק? מי יטפל בבעיות?
6. שינוי ארגוני: התנגדות של עובדים לשינוי

## 3.1 הרכבת צוות IA

### 1.3.1 תפקידים בצוות IA

צוות IA מוצלח הוא רב-תחומי [5]. להלן התפקידים המרכזיים:

דיקפת	תורחא	נמז %
AI Product Manager	יינע ילעב סע תרושקת, פודעת, תושירד תרדגה	100%
Data Scientist	Prompt Engineering, סילדומ תריחב, רקחמ	60-100%
ML Engineer	MLOps, היצרגטניא, חותיפ	100%
Data Engineer	ETL, Vector DB, סינותנ תיתשת	40-60%
Backend Developer	הסירפ, היצרגטניא, API	60%
Frontend Developer	UX, שמתשמ קשממ	40-60%
DevOps Engineer	רוטינ, CI/CD, תיתשת	40%
QA Engineer	תוקידב, Evaluation, פוקית	60%

טבלה 1.1: תפקידים בצוות IA טיפוי

לא כל ארגון יכול להרשות לעצמו צוות של 8 אנשים. לפרויקט קטן, די ב:

□ reganaM tcudorP IA 1

□ repoleveD kcatS-lluF 1 עם ידע בסיסי ב-IA

□ reenignE LM/tsitneicS ataD 1 (יכול להיות חלקי)

שלושת אנשים אלה יכולים להוציא לפועל COP ו-toliP. כשמרחיבים לייצור -- תגייסו עוד.

### 2.3.1 בניית הצוות -- rentraP sv yuB sv dliuB

יש שלוש אסטרטגיות לבניית צוות IA [61]:

1. dliuB (בניה פנימית):

□ יתרונות: שליטה מלאה, ידע נשאר בארגון

□ חסרונות: גיוס קשה, עלות גבוהה

□ מתי: כשיש תקציב ופרויקטים ארוכי טווח

## 2. yuB (השכרת חברה חיצונית):

- יתרונות: מומחיות מיידית, ללא גיוס
- חסרונות: עלות גבוהה, תלות בספק
- מתי: לפרויקטים חד-פעמיים או COP

## 3. rentraP (שותפות היברידית):

- יתרונות: העברת ידע, בניית יכולת פנימית
- חסרונות: דורש תיאום, לוקח זמן
- מתי: המודל המומלץ ברוב המקרים

**המלצה:** התחילו עם rentraP (צוות חיצוני + 1-2 אנשים פנימיים), ובנו בהדרגה צוות פנימי.

## 4.1 ניהול פרויקט IA בגישת eligA

## 1.4.1 למה eligA מתאים ל-IA?

פרויקט IA הם בעלי אי-ודאות גבוהה [7]. אתה לא יודע מראש אם מודל מסוים יעבוד, או איך המשתמשים יגיבו. לכן, גישת IlafretaW המסורתית (תכנון מלא מראש, פיתוח ארוך, בדיקה בסוף) אינה מתאימה.

eligA מציעה [1], [12]:

- איטרציות קצרות (stnirpS): כל 2 שבועות יש תוצר עובד
- משוב מהיר: בודקים עם משתמשים כל הזמן
- גמישות: אפשר לשנות כיוון על סמך למידה
- שיתוף פעולה: צוות עובד יחד, לא בסילואים

## 2.4.1 מבנה tnirpS ב-IA tcejorP

tnirpS טיפוסי באורך שבועיים:

סוי	תוליעפ	סיפתתשמ
סוי 1 (ב')	Sprint Planning -- ההזדמנות ב-Sprint הנבחר	תוצה לכ
סוי 2-9	Daily Standups (15 תוקד) + חותיפ תדובע	תוצה לכ
סוי 10 (ה')	Sprint Demo -- יינע ילעבל רצותה תגצה	Stakeholders + תווצ
סוי 10 (ה')**	Retrospective -- דבע המ	תוצה לכ

טבלה 2.1: מבנה tnirpS דו-שבועי

**\*\* omeD ו-orteR באותו יום, אחד אחרי השני**

### 3.4.1 IA-seirotS resU ל-IA

ב-eligA, כל דרישה מנוסחת כ-"yrotS resU" [4].  
**תבנית:**

"כ-[תפקיד], אני רוצה [פעולה], כדי [ערך עסקי]"

#### דוגמאות:

□ "כנציג שירות, אני רוצה שהצ'אטבוט ימליץ לי על מוצר מתאים ללקוח, כדי לשפר את שביעות הרצון"

□ "כמנהל RH, אני רוצה לקבל רשימת 02 מועמדים מובילים לכל משרה, כדי לחסוך זמן סינון"

□ "כמנהל, אני רוצה דאשבורד עם sIPK של מערכת ה-IA, כדי לעקוב אחר ביצועים"  
לכל yrotS יש:

□ `airetirC ecnatpecca`: מתי ה-yrotS נחשבת גמורה?

□ `stnioP yrotS`: הערכת מאמץ (1, 2, 3, 5, 8, ...31)

### 4.4.1 tnmeganaM golkcaB 4.4.1

ה-`golkcaB tcudorP` הוא רשימת כל ה-seirotS. ה-IA `reganaM tcudorP` אחראי:

1. לתחזק את ה-`golkcaB` (להוסיף, למחוק, לעדכן)

2. לתעדף לפי ערך עסקי

3. להבטיח שה-seirotS ברורות ומובנות

בלחץ להוציא תכונות, צוותים לפעמים "מקצרים דרך" ולא עושים `gnitset`, `gnirotcafeR`, `noitatnemucoD`. זה נקרא `tbeD lacinhceT` [22].  
**הכלל:** הקדישו 02% מכל `tnirpS` לתשלום "חוב טכני" -- `stseT`, `gnirotcafeR`, ביצועים, תיעוד.

## 5.1 יסודות spOLM -- תחזוקה ושיפור מתמיד

### 1.5.1 מהו spOLM?

spOLM (`gninraeL enihcaM` `snoitarepO`) הוא `spOveD` עבור מערכות IA [11], [42]. מטרתו: להבטיח שמערכות IA רצות באופן יציב, מתוחזקות ומשתפרות לאורך זמן.  
**רכיבי spOLM:**

1. `lortnoC noisreV`: גרסה של קוד, מודלים, `stpmorP`, נתונים

2. `DC/IC`: אינטגרציה ופריסה אוטומטיות

3. `gnirotinoM`: ניטור ביצועים, שגיאות, שימוש

4. `gniggoL`: רישום אירועים לניתוח

5. `gnitreIA`: התרעות כשמשו לא תקין

6. `gniniarteR`: עדכון מודלים/`stpmorP` לפי צורך

## 2.5.1 lortnoC noisreV למערכות IA

בפיתוח תוכנה רגיל, אתה עושה lortnoC noisreV לקוד בלבד. במערכות IA, צריך גם [52], [72]:

- gninoisreV ledom: כל שינוי במודל (sretemaraP, tpmorP) מתועד
- gninoisreV ataD: מעקב אחר גרסאות של stesatad
- gnikcarT tnemirepxE: תיעוד של כל ניסוי (מודל, tpmorP, תוצאות)

כלים:

- tiG לקוד
- CVD (lortnoC noisreV ataD) לנתונים
- sesaiB & sthgieW / wolFLM לניסויים

## 3.5.1 (DC/IC) tnemyolpeD suounitnoC / noitargetnI suounitnoC

IC (noitargetnI suounitnoC): כל שינוי בקוד עובר אוטומטית [9], [01]:

1. dliuB (בניית הקוד)
  2. stseT (בדיקות אוטומטיות)
  3. skcehC ytilauQ (gnitniL, ytiruceS, nacS)
- DC (tnemyolpeD suounitnoC): אם הכל עובר, הקוד נפרס אוטומטית לסביבת ייצור.  
enilepiP טיפוס:

edoC hsuP edoC buHtiG DC/IC (snoitcA buHtiG) egamI rekcoD dliuB stseT nuR duolC ot yolpeD

## 4.5.1 gniggoL-ו gnirotinoM

מה לנטר? [3], [41]

- ביצועי מודל: ycnetaL (זמן תגובה), tuphguorhT (בקשות לדקה)
- איכות תשובות: ecnaveleR, ycaruccA (באמצעות noitaulavE)
- שגיאות: sepyT noitpecxE, etaR rorrE
- שימוש: כמה משתמשים, כמה שאילתות, kaeP sruoH
- עלויות: כמה טוקנים, כמה זה עולה

כלים:

- anafarG + suehtemorP: ניטור ודאשבורדים
- kcatS KLE (anabiK, hsatsgoL, hcraescitsale): gniggoL וחיפוש
- yrtneS: מעקב שגיאות
- enocileH / htimSgnal: ניטור ספציפי ל-MLL



### 5.5.1 gniniarteR-ב והצורך tfirD ledom

בזמן, ביצועי מערכות IA יכולים להידרדר. זה נקרא tfirD ledom [8], [21].  
**למה זה קורה?**

□ הנתונים משתנים (למשל, לקוחות שואלים שאלות חדשות)

□ המציאות משתנה (מוצרים חדשים, מחירים)

□ stpmorP שפעם עבדו כבר לא מספיק טובים

**פתרון:** gniniarteR / etadpU tpmorP

1. ניטור ביצועים קבוע

2. כשאיכות יורדת מתחת לסף -- התרעה

3. עדכון stpmorP / gninut-eniF / החלפת מודל

4. בדיקה ופריסה מחדש

## 6.1 מדידת הצלחה -- sIPK למערכות IA

### 1.6.1 למה sIPK קריטיים?

"מה שלא נמדד, לא מנוהל" -- פיטר דראקר  
בלי מדידה, אתה לא יודע אם הפרויקט הצליח, איפה לשפר, או אם להמשיך להשקיע. sIPK  
הם המצפן שלך [81].

### 2.6.1 שלושה סוגי sIPK

1. sIPK עסקיים (sIPK ssenisuB): מודדים ערך עסקי ישיר [6], [91].

$$AI\_Project\_ROI = \frac{\text{תללוק העקשה} - \text{רצון דרע}}{\text{תללוק העקשה}} \times 100\%$$

**דוגמה:**

□ ערך נוצר: 000,005 ש"ח (חיסכון שנתי בעלויות שירות)

□ השקעה: 000,002 ש"ח (פיתוח + תחזוקה שנה ראשונה)

□  $IOR = 000,002 / (000,002 - 000,005) = 051\%$

**sIPK עסקיים נוספים:**

□ **חיסכון בעלויות:** כמה כסף חסכנו?

□ **הגדלת הכנסות:** כמה הכנסות נוספות הניבה המערכת?

□ **שיפור שביעות רצון לקוחות (TASC):** לפני ואחרי

□ eulaV ot emiT: כמה זמן עד שראינו ערך ראשון?

(סישדוח/תועובשב) First Value דע Kickoff-מ ןמז = Time\_to\_Value

**דוגמה:**

□ ffokciK: 1 בינואר

□ eulaV tsriF (COP הוכיח ערך): 51 בפברואר

□ eulaV ot emiT = 6 שבועות

**הערה:** ככל ש-VTT קצר יותר, כך הפרויקט מהיר יותר בהניבת ערך.

**2. sIPK טכניים (sIPK lacinheT):** מודדים ביצועים טכניים.

□ ycnetaL: זמן תגובה ממוצע (בשניות)

□ tuphguorhT: בקשות לדקה

□ etaR rorrE: % שגיאות

□ emitpU: % זמן זמינות (ALS: 99.99%)

□ egasU nekoT: צריכת טוקנים ממוצעת לבקשה

**3. sIPK של אימוץ (sIPK noitpodA):** מודדים עד כמה המשתמשים משתמשים במערכת.

$$\text{Adoption\_Rate} = \frac{\text{סילעפ שישמש}}{\text{סיילאצנטופ שישמש}} \times 100\%$$

**דוגמה:**

□ משתמשים פוטנציאליים (כל נציגי השירות): 05

□ משתמשים פעילים (השתמשו לפחות פעם בשבוע): 04

□ etaR noitpodA = 05/04 = 08%

**sIPK אימוץ נוספים:**

□ sresU evitcA yliaD (UAD): כמה משתמשים ביום

□ etaR tnemegagnE: % מהמשתמשים שחוזרים

□ egasU erutaeF: איזה תכונות בשימוש ואיזה לא

### 3.6.1 בניית draobhsaD ל-sIPK

draobhsaD טוב מציג את כל ה-sIPK במקום אחד, בזמן אמת.  
מבנה draobhsaD מומלץ:

#### 1. סקירה כללית (weivrevO):

IOR □

etaR noitpodA □

emitpU □

#### 2. ביצועים טכניים:

ycnetaL לאורך זמן □

etaR rorrE □

egasU nekoT □

#### 3. שימוש:

(sresU evitcA yliaD) UAD □

sruoH kaeP □

תכונות פופולריות □

#### 4. עלויות:

עלות IPA חודשית □

עלות לבקשה □

תחזית עלויות □

#### כלים לבניית draobhsaD:

anafarG: פתוח, חזק, מקצועי □

tilmaertS: פשוט, מהיר, desab-nohtyP □

IB rewoP / uaelbaT: עסקי, חזותי □

## 7.1 דוגמאות מעשיות

### 1.7.1 1: ydutS esaC פרויקט IA מוצלח מ-A עד Z

ארגון: בנק בינוני עם 002 עובדים

אתגר: תהליך אישור הלוואות לעסקים קטנים אורך 2-3 שבועות ודורש עבודה ידנית רבה של אנליסטים.

yrevoCSI (שבועיים):

ריאיון עם 01 אנליסטים □

מיפוי תהליך אישור הלוואות (41 שלבים!) □

זיהוי: 6 שלבים ניתנים לאוטומציה עם IA □

□ ערך פוטנציאלי: חיסכון 06% בזמן

#### COP (3 שבועות):

□ בניית סוכן IA שמנתח מסמכים פינסיים (דוחות רווח והפסד, מאזנים)

□ מחלץ נתונים מרכזיים ומייצר "ציון סיכון" אוטומטי

□ נבדק על 02 הלוואות קודמות

□ תוצאה: דיוק 58% בהשוואה לאנליסטים אנושיים

#### tolIP (חודשיים):

□ הרחבה ל-001 הלוואות חדשות

□ 5 אנליסטים משתמשים במערכת

□ זמן אישור ממוצע ירד מ-41 ימים ל-7 ימים

□ משוב: המערכת טובה, אך צריכה שיפור בזיהוי מסמכים סרוקים

#### noitcudorP (לאחר 4 חודשים):

□ השקה מלאה לכל מחלקת ההלוואות

□ 001% מהלוואות עוברות דרך המערכת

□ זמן אישור ממוצע: 5 ימים (שיפור 46%)

□ חיסכון שנתי: 2.1 מיליון ש"ח

□ IOR: 004% בשנה הראשונה

#### לקחים:

□ yrevocsiD מסודר חוסך זמן בהמשך

□ COP קצר עם מדדים ברורים מקטין סיכונים

□ שיתוף משתמשי קצה מוקדם מבטיח הצלחה

□ שיפורים על בסיס משוב toliP קריטיים

### 2.7.1 ydutS esaC 2: ניתוח כישלון -- מה לא לעשות

**ארגון:** חברת ביטוח גדולה

**רעיון:** בניית "סוכן IA אוניברסלי" שיטפל בכל פניות הלקוחות.

**מה השתבש?**

**1. yrevocsiD חסר:**

□ ההנהלה החליטה לבנות IA בלי לשאול את נציגי השירות

□ לא זיהו את הכאבים האמיתיים

□ תוצאה: פתרון לבעיה שלא הייתה

**2. אין COP -- ישר לפיתוח מלא:**

- החליטו לפתח מערכת ענקית של 6 חודשים
- לא בדקו היתכנות טכנולוגית
- תוצאה: גילו רק בסוף שהמודל לא מספיק טוב

**3. צוות לא מתאים:**

- שכרו חברה חיצונית יקרה בלי ידע פנימי
- לא העבירו ידע לארגון
- תוצאה: תלות מוחלטת בספק

**4. אין sIPK:**

- לא הגדירו מדדי הצלחה
- לא ניטרו ביצועים בזמן הפיתוח
- תוצאה: גילו רק בסוף שהמערכת לא עובדת

**5. peerC epocS:**

- כל פעם הוסיפו דרישות חדשות
- הפרויקט התנפח מ-6 חודשים ל-81 חודשים
- תוצאה: תקציב התפוצץ (5.2 מיליון במקום 000,008)

**תוצאה סופית:**

- המערכת נזנחה לאחר 81 חודשים
- הפסד 5.2 מיליון ש"ח
- תדמית IA בארגון נפגעה
- לקח 3 שנים עד שנכוננו לנסות שוב

**לקחים:**

- **אל תדלגו על yrevocsiD** -- זה לא בזבז זמן, זה חיסכון
- **תתחילו ב-COP קטן** -- הוכיחו ערך לפני השקעה גדולה
- **בנו ידע פנימי** -- אל תהיו תלויים לחלוטין בספקים
- **הגדירו sIPK מראש** -- תדעו מתי להמשיך ומתי לעצור
- **שמרו על epocS קבוע** -- תגידו "לא" לדרישות חדשות

**3.7.1 תכנון pamdaoR למחלקת IA****pamdaoR ל-21 חודשים:****רבעון 1 (חודשים 1-3):**

□ sesaC esU 5 זיהוי + yrevocsiD

□ COP לשני sesaC esU הטובים ביותר

□ בניית צוות ליבה (3 אנשים)

□ הגדרת sIPK

**רבעון 2 (חודשים 4-6):**

□ toliP ל-esU sesaC הראשון עם 02 משתמשים

□ הרחבת הצוות (2 אנשים נוספים)

□ בניית תשתית spOLM בסיסית

□ הדרכת משתמשים

**רבעון 3 (חודשים 7-9):**

□ oG eviL ל-esU sesaC הראשון

□ toliP ל-esU sesaC השני

□ בניית draobhsaD ל-sIPK

□ תיעוד ושיתוף לקחים

**רבעון 4 (חודשים 01-21):**

□ oG eviL ל-esU sesaC השני

□ COP ל-esU sesaC השלישי

□ הערכת IOR של שנה ראשונה

□ תכנון pamdaoR לשנה השנייה

**התוצאה לאחר שנה:**

□ 2 מערכות IA בייצור

□ צוות של 5 אנשים עם ידע מעמיק

□ IOR מוכח

□ תשתית spOLM מוכנה להרחבה

□ תרבות IA מתחילה להיבנות בארגון

## 8.1 תרגילים

### 1.8.1 תרגיל 1: כתיבת retrahC tcejorP לפרויקט IA

retrahC tcejorP הוא מסמך הפתיחה של פרויקט. הוא מגדיר מה, למה, מי ומתי.  
**משימה:** כתבו retrahC tcejorP לפרויקט IA בארגון שלכם (או בארגון דמיוני).  
**מבנה retrahC tcejorP:**

1. שם הפרויקט
2. רקע ומטרה: למה אנחנו עושים את זה?
3. היקף (epocS): מה בפנים, מה בחוץ
4. מטרות (slaoG): מדידות וספציפיות
5. sIPK: איך נמדוד הצלחה?
6. בעלי עניין (sredlohekatS): מי מעורב?
7. צוות: מי עובד על הפרויקט?
8. לוח זמנים: מתי זה יגמר?
9. תקציב: כמה זה יעלה?
10. סיכונים: מה יכול להשתבש?

דוגמה:

1. שם הפרויקט: metsyS gnineercS VC derewoP-IA
2. רקע ומטרה: מחלקת RH מבלה כ-04 שעות בחודש על סינון ידני של קורות חיים. המטרה: להפחית זמן סינון ב-07% באמצעות IA.
3. היקף:

□ **בפנים:** סינון קורות חיים, דירוג מועמדים, המלצת pot 02

□ **בחוץ:** ריאיון אוטומטי, הצעת עבודה אוטומטית

4. מטרות:

□ הפחתת זמן סינון מ-04 שעות ל-21 שעות בחודש

□ שיפור איכות מועמדים שעוברים לראיון ב-02%

□ eviL oG תוך 3 חודשים

5. sIPK:

□ זמן סינון ממוצע למשרה

□ % מועמדים שעברו לשלב ראיון והתקבלו

□ etaR noitpodA (כמה משתמשי RH משתמשים)

□ IOR לאחר 6 חודשים

#### 6. בעלי עניין:

□ מנהלת RH (rosnopS)

□ צוות sretiurceR (משתמשי קצה)

□ OTC (תומך טכני)

#### 7. צוות:

□ reganaM tcudorP (חצי משרה)

□ reenignE LM (משרה מלאה)

□ reenignE AQ (רבע משרה)

#### 8. לוח זמנים:

□ COP: 3 שבועות

□ toliP: 4 שבועות

□ eviL oG: 3 חודשים מהיום

#### 9. תקציב:

□ שכר צוות: 000,051 ש"ח

□ stsoC IPA: 000,5 ש"ח

□ כלים: 000,01 ש"ח

□ סה"כ: 000,561 ש"ח

#### 01. סיכונים:

□ קושי בגישה לנתוני קורות חיים (הסתברות: בינונית)

□ התנגדות של sretiurceR (הסתברות: נמוכה)

□ דיוק המודל נמוך מ-08% (הסתברות: בינונית)

### 2.8.1 תרגיל 2: תכנון צוות IA אידיאלי

**משימה:** תכננו את הצוות האידיאלי לפרויקט IA בארגון שלכם.  
**שאלות מנחות:**

1. מהו גודל הפרויקט? (קטן / בינוני / גדול)

2. מהו התקציב?

3. האם יש ידע פנימי ב-IA?

4. האם עדיף dliuB / yuB / rentraP?



5. אילו תפקידים נדרשים?

6. כמה זמן (חלקי/מלא) לכל תפקיד?

**פלט מצופה:** טבלה עם:

- שם תפקיד
- אחריות
- % זמן
- פנימי / חיצוני
- עלות חודשית

### 3.8.1 תרגיל 3: בניית draobhsaD של sIPK

**משימה:** עצבו draobhsaD ויזואלי (על נייר או כלי עיצוב) שמציג sIPK למערכת IA. **דרישות:**

- 4 אזורים: עסקי, טכני, אימוץ, עלויות
- לפחות 3 מדדים בכל אזור
- כלול גרפים (עמודות, קו, עוגה)
- סימנו אזורים קריטיים באדום/ירוק

**כלים מומלצים:**

- amgiF / hctekS (עיצוב)
- lecxE / elgooG / steehS (מוקאפ מהיר)
- anafarG (אם יש זמן ללמוד)

### 4.8.1 תרגיל 4: ניתוח ydutS esaC ולמידת לקחים

**משימה:** קראו את שני ה-seidutS esaC (הצלחה וכישלון) ומלאו טבלה:

אשון	החלצה Case	ולשיכ Case
Discovery	תועובש 2, רדוסמ	היה אל
POC	סירורב סידדמ, תועובש 3	וילע וגליד
תווצ	...	...
KPIs	...	...
Scope	...	...
האצות	ROI 400%	2.5M דספה

טבלה 3.1: השוואת seidutS esaC

**שאלות להרחבה:**

1. מה היה הגורם המרכזי להצלחה ב-esac הראשון?
2. מה היה הגורם המרכזי לכישלון ב-esac השני?
3. אילו לקחים תיקחו לפרויקט שלכם?

**5.8.1 תרגיל 5: תכנון pamdaoR IA ל-21 חודשים**

**משימה:** תכננו pamdaoR של 21 חודשים לפרויקט IA בארגון שלכם.  
**מבנה:**

□ חלקו ל-4 רבעונים

□ כל רבעון: מטרות, פעילויות, selbarevileD, משאבים

**דוגמה לרבעון 1:**

□ **מטרות:** זיהוי הזדמנויות, COP

□ **פעילויות:** yrevocsiD, COP, בניית צוות

□ **selbarevileD:** מסמך paM ytinutroppO, COP omeD

□ **משאבים:** 3 אנשים, K001 ש"ח

**6.8.1 תרגיל 6 (קוד nohtyP): מערכת ניטור sIPK בסיסית**

**משימה:** כתבו סקריפט nohtyP שמחשב ומציג sIPK בסיסיים למערכת IA.  
**נתונים לדוגמה (NOSJ):**

```
1 {
2   "total_requests": 10000,
3   "successful_requests": 9500,
4   "failed_requests": 500,
5   "total_tokens": 5000000,
6   "total_cost_usd": 250,
7   "potential_users": 100,
8   "active_users": 75,
9   "value_generated_usd": 10000,
10  "total_investment_usd": 5000
11 }
```

**sIPK לחישוב:**

1.  $\text{stseuqer\_latot} / \text{stseuqer\_lufsseccus} = \text{etaR sseccuS}$  001

2.  $\text{stseuqer\_latot} / \text{tsoc\_latot} = \text{tseuqeR rep tsoC}$

3.  $\text{sresu\_laitnetop} / \text{sresu\_evitca} = \text{etaR noitpodA}$  001

4.  $\text{tnemtsevni\_latot} / (\text{tnemtsevni\_latot} - \text{detareneg\_eulav}) = \text{IOR}$  001

5.  $\text{stseuqer\_latot} / \text{snekoT\_latot} = \text{tseuqeR rep snekoT egarevA}$

**פתרון:**

Listing 1.1: kpi\_monitor.py

```
1 import json
2
3 def calculate_kpis(data):
4     """
```

```
5 Calculate basic KPIs for AI system
6
7 Args:
8     data (dict): System usage data
9
10 Returns:
11     dict: Calculated KPIs
12 """
13 kpis = {}
14
15 # Success Rate
16 if data['total_requests'] > 0:
17     kpis['success_rate'] = (
18         data['successful_requests'] /
19         data['total_requests'] * 100
20     )
21 else:
22     kpis['success_rate'] = 0
23
24 # Cost per Request
25 if data['total_requests'] > 0:
26     kpis['cost_per_request'] = (
27         data['total_cost_usd'] /
28         data['total_requests']
29     )
30 else:
31     kpis['cost_per_request'] = 0
32
33 # Adoption Rate
34 if data['potential_users'] > 0:
35     kpis['adoption_rate'] = (
36         data['active_users'] /
37         data['potential_users'] * 100
38     )
39 else:
40     kpis['adoption_rate'] = 0
41
42 # ROI
43 if data['total_investment_usd'] > 0:
44     kpis['roi'] = (
45         (data['value_generated_usd'] -
46          data['total_investment_usd']) /
47         data['total_investment_usd'] * 100
48     )
49 else:
50     kpis['roi'] = 0
51
52 # Average Tokens per Request
53 if data['total_requests'] > 0:
54     kpis['avg_tokens_per_request'] = (
55         data['total_tokens'] /
```

```

56         data['total_requests']
57     )
58     else:
59         kpis['avg_tokens_per_request'] = 0
60
61     return kpis
62
63 def display_kpis(kpis):
64     """Display KPIs in clean format"""
65     print("=" * 50)
66     print("AI System KPIs Dashboard")
67     print("=" * 50)
68     print(f"Success Rate: {kpis['success_rate']:.2f}%")
69     print(f"Cost per Request: ${kpis['cost_per_request']:.4f}")
70     print(f"Adoption Rate: {kpis['adoption_rate']:.2f}%")
71     print(f"ROI: {kpis['roi']:.2f}%")
72     print(f"Avg Tokens/Request: {kpis['avg_tokens_per_request']:.0f
73 }")
74     print("=" * 50)
75
76 def main():
77     # Sample data
78     data = {
79         "total_requests": 10000,
80         "successful_requests": 9500,
81         "failed_requests": 500,
82         "total_tokens": 5000000,
83         "total_cost_usd": 250,
84         "potential_users": 100,
85         "active_users": 75,
86         "value_generated_usd": 10000,
87         "total_investment_usd": 5000
88     }
89
90     # Calculate and display KPIs
91     kpis = calculate_kpis(data)
92     display_kpis(kpis)
93
94 if __name__ == "__main__":
95     main()

```

פלט מצופה:

```

=====
                        draobhsaD sIPK metsyS IA
=====
                        %00.59 :etaR sseccuS
                        0520.0$ :tseuger rep tsoC
                        %00.57 :etaR noitpodA
                        %00.001 :IOR
                        005 :tseuger/snekoT gvA
=====

```

## שיפורים אפשריים:

1. הוספת צבעים (ירוק/אדום) לפי סף
2. שמירת היסטוריה ב-VSC
3. גרפים עם biltolptam
4. התרעות כש-IPK יורד מתחת לסף

### 7.8.1 תרגיל 7 (קוד nohtyP): gniggoL ו-gnirotinoM למערכת IA

**משימה:** בנו מערכת gniggoL בסיסית שמתעדת כל שימוש במערכת IA ושומרת goL eliF.  
**מה לתעד?**

pmatsemiT □  
DI resU □  
(tpmorP) tseuqeR □  
esnopseR □  
desU snekoT □  
(sm) ycnetaL □  
rorrE / sseccuS □

## פתרון:

Listing 1.2: ai\_logger.py

```
1 import json
2 import logging
3 from datetime import datetime
4 from pathlib import Path
5
6 class AISystemLogger:
7     """Logging system for AI applications"""
8
9     def __init__(self, log_dir="logs"):
10         """Initialize Logger
11
12         Args:
13             log_dir (str): Logs directory
14         """
15         self.log_dir = Path(log_dir)
16         self.log_dir.mkdir(exist_ok=True)
17
18         # Configure Logger
19         self.logger = logging.getLogger("AISystem")
20         self.logger.setLevel(logging.INFO)
21
22         # File Handler
```

```

23 |     log_file = self.log_dir / f"ai_system_{datetime.now().
    strftime('%Y%m%d')}.log"
24 |     file_handler = logging.FileHandler(log_file, encoding='utf
    -8')
25 |     file_handler.setLevel(logging.INFO)
26 |
27 |     # Format
28 |     formatter = logging.Formatter(
29 |         '%(asctime)s - %(name)s - %(levelname)s - %(message)s'
30 |     )
31 |     file_handler.setFormatter(formatter)
32 |
33 |     self.logger.addHandler(file_handler)
34 |
35 |     def log_request(self, user_id, prompt, response,
36 |                     tokens_used, latency_ms, success=True, error_msg
    =None):
37 |         """Log AI system request
38 |
39 |         Args:
40 |             user_id (str): User identifier
41 |             prompt (str): The prompt sent
42 |             response (str): The response received
43 |             tokens_used (int): Token count
44 |             latency_ms (int): Response time in milliseconds
45 |             success (bool): Whether request succeeded
46 |             error_msg (str): Error message (if any)
47 |         """
48 |         log_entry = {
49 |             "timestamp": datetime.now().isoformat(),
50 |             "user_id": user_id,
51 |             "prompt": prompt[:100], # First 100 chars only
52 |             "response": response[:100] if response else None,
53 |             "tokens_used": tokens_used,
54 |             "latency_ms": latency_ms,
55 |             "success": success,
56 |             "error_msg": error_msg
57 |         }
58 |
59 |         if success:
60 |             self.logger.info(json.dumps(log_entry, ensure_ascii=
    False))
61 |         else:
62 |             self.logger.error(json.dumps(log_entry, ensure_ascii=
    False))
63 |
64 |     def get_daily_stats(self):
65 |         """Calculate daily statistics from log file
66 |
67 |         Returns:
68 |             dict: Statistics

```

```

69         """
70         log_file = self.log_dir / f"ai_system_{datetime.now().
strftime('%Y%m%d')}.log"
71
72         if not log_file.exists():
73             return {"error": "No log file for today"}
74
75         total_requests = 0
76         successful_requests = 0
77         failed_requests = 0
78         total_tokens = 0
79         total_latency = 0
80
81         with open(log_file, 'r', encoding='utf-8') as f:
82             for line in f:
83                 if 'prompt' in line:
84                     total_requests += 1
85
86                 # Extract JSON from line
87                 json_start = line.find('{')
88                 if json_start != -1:
89                     try:
90                         entry = json.loads(line[json_start:])
91
92                         if entry.get('success', False):
93                             successful_requests += 1
94                         else:
95                             failed_requests += 1
96
97                         total_tokens += entry.get('tokens_used'
, 0)
98                         total_latency += entry.get('latency_ms'
, 0)
99
100                     except json.JSONDecodeError:
101                         continue
102
103         stats = {
104             "date": datetime.now().strftime('%Y-%m-%d'),
105             "total_requests": total_requests,
106             "successful_requests": successful_requests,
107             "failed_requests": failed_requests,
108             "success_rate": (successful_requests / total_requests *
100)
109
110             if total_requests > 0 else 0,
111             "total_tokens": total_tokens,
112             "avg_tokens_per_request": (total_tokens /
total_requests)
113
114             if total_requests > 0 else 0,
115             "avg_latency_ms": (total_latency / total_requests)
116
117             if total_requests > 0 else 0

```

```

115     }
116
117     return stats
118
119 # Usage example
120 def main():
121     # Initialize Logger
122     logger = AISystemLogger()
123
124     # Simulate requests
125     logger.log_request(
126         user_id="user_123",
127         prompt="What is GPT-4?",
128         response="GPT-4 is a large language model...",
129         tokens_used=150,
130         latency_ms=1200,
131         success=True
132     )
133
134     logger.log_request(
135         user_id="user_456",
136         prompt="Calculate something complex",
137         response=None,
138         tokens_used=50,
139         latency_ms=500,
140         success=False,
141         error_msg="Timeout"
142     )
143
144     logger.log_request(
145         user_id="user_789",
146         prompt="Write me an email",
147         response="Hello, I am writing to you...",
148         tokens_used=200,
149         latency_ms=1500,
150         success=True
151     )
152
153     # Display daily statistics
154     stats = logger.get_daily_stats()
155     print("\nDaily Statistics:")
156     print("=" * 50)
157     for key, value in stats.items():
158         if isinstance(value, float):
159             print(f"{key}: {value:.2f}")
160         else:
161             print(f"{key}: {value}")
162     print("=" * 50)
163
164 if __name__ == "__main__":
165     main()

```



## פלט מצופה:

```

:scitsitatS yliaD
=====
51-10-5202 :etad
3 :stseuqer_latot
2 :stseuqer_lufsseccus
1 :stseuqer_deliaf
76.66 :etar_sseccus
004 :snekot_latot
33.331 :tseuqer_rep_snekot_gva
76.6601 :sm_ycnetal_gva
=====

```

## שיפורים אפשריים:

1. שליחת התרעות (kcalS/liamE) כשיש שגיאה

2. אחסון ב-esabataD במקום קובץ

3. draobhsaD בזמן אמת עם tilmaertS

4. ניתוח טרנדים לאורך זמן

## 9.1 סיכום

פרק זה לקח אתכם למסע המלא -- מזיהוי הזדמנויות IA בארגון, דרך COP ו-toliP, ועד לייצור מלא ותחזוקה מתמדת. למדנו כיצד להרכיב צוות IA, לנהל פרויקט בגישת eligA, ליישם עקרונות spOLM, ולמדוד הצלחה באמצעות sIPK.

### נקודות מפתח:

1. yrevocsiD קפדני חוסך זמן וכסף -- אל תדלגו עליו

2. COP קטן ומהיר עדיף על פרויקט ענק ארוך

3. צוות היברידי (פנימי + חיצוני) הוא לרוב הפתרון הטוב ביותר

4. eligA מתאים ל-IA -- איטרציות קצרות ומשוב מהיר

5. spOLM הוא הכרח -- בלעדיו המערכת תקרוס

6. sIPK הם המצפן -- מה שלא נמדד, לא מנוהל

בסופו של דבר, הצלחה בפרויקט IA היא לא עניין של טכנולוגיה בלבד. היא עניין של ניהול נכון, תכנון מושכל, צוות טוב, ומדידה מתמדת. כמו תוכנית אפולו ב-9691 -- הטכנולוגיה חשובה, אבל הניהול הוא שעושה את ההבדל.

אתם עכשיו מצוידים בידע ובכלים להוביל פרויקט IA מוצלח בארגון שלכם. זכרו: התחילו קטן, הוכיחו ערך, למדו, שפרו, והרחיבו. בהצלחה במסע!

## מקורות והמלצות לקריאה נוספת

1. koohS nhoJ -- "koobyalP IA naeL ehT" :IA rof eligA
2. [42] .la te lieverT kraM -- "spOLM gnicudortnl" :spOLM
3. [02] esoR guoD -- "ssenisuB rof IA" :tnemeganaM tcejorP IA
4. [81] retnemraP divaD -- "srotacidnI ecnamrofreP yeK" :sIPK
5. [32] tropeR xednI IA IAH drofnatS :seidutS esaC

## מקורות

- [1] ,gninnerG .J ,relwoF .M ,mahgninnuC .W ,nrubkcoC .A ,mukenneB nav .A ,eldeeB .M ,kceB .K .J ,rebawhcS .K ,rolleM .S ,nitram .C .R ,kciraM .B ,nreK .J ,seirffeJ .R ,tnuH .A ,htimshgiH .J ,1002 ,ecnaillA eligA .tnempoleveD erawtfoS eligA rof otsefinaM ,samohT .D dna ,dnalrehtuS <https://agilemanifesto.org/> .:elbaliavA .[enilnO] .otsefinaM eligA lanigiro ehT
- [2] " ,adnegA hcraeseR A :smetsyS IA gnireenignE` ,civoknrC .I dna ,nossIO .H .H ,hcsob .J .pp ,91-1 ,0202 :IOD .10.4018/978-1-7998-5101-1.ch001
- [3] cirbuR A :erocS tseT LM ehT` ,yellucS .D dna ,bilaS .M ,nesleiN .E ,iaC .S ,kcerB .E :IOD .7102 ,2311-3211 .pp " ,noitcudeR tbeD lacinheC dna ssenidaeR noitcudorP LM rof [10.1109/BigData.2017.8258038](https://doi.org/10.1109/BigData.2017.8258038)
- [4] ,lanoisseforP yelseW-nosiddA .tnempoleveD erawtfoS eligA rof :deilppA seirotS resU ,nhoC .M .1865021230-879 :NBSI ,4002
- [5] .E dna ,ovla□eP-a□craG .J .F ,atsocA-otoS .P ,sarerbmuL-odasaC .C ,soicalaP-omoloC .R " ,yduTS lanoitazinagro-itluM A :lennosreP erawtfoS ni spaG ecnetepmoC` ,oraC-ravoT [10.1016/j.chb.2012.04.021](https://doi.org/10.1016/j.chb.2012.04.021) :IOD .3102 ,164-654 .pp ,2 .on ,92 .lov ,roivaheB namuH ni sretupmoC
- [6] ,69 .lov ,8102 .dlroW laeR eht rof ecnegilletnI laicifitra ,iknanaR .R dna tropnevaD .H .T .seigetarts noitatnemelpmi IA lacitcarP ,611-801 .pp
- [7] :seigolodohteM eligA fo edaceD A` ,eoM .B .N dna ,yllapejilaB .V ,rureN .S ,ry□sgniD .T ,58 .lov ,erawtfoS dna smetsyS fo lanuoJ " ,tnempoleveD erawtfoS eligA gninialpxE sdrawoT [10.1016/j.jss.2012.02.033](https://doi.org/10.1016/j.jss.2012.02.033) :IOD .2102 ,1221-3121 .pp ,6 .on
- [8] tpecnoC no yevruS A` ,aihcahuoB .A dna ,yikzinehceP .M ,tefiB .A ,□tiaboil□ .I ,amaG .J [10.1145/2523813](https://doi.org/10.1145/2523813) :IOD .4102 ,73-1 .pp ,4 .on ,64 .lov ,syevruS gnitupmoC MCA " ,noitatpadA tfirD
- [9] ,dliuB hguorht sesaeleR erawtfoS elbaileR :yrevileD suounitnoC ,yelraF .D dna elbmuH .J .9191061230-879 :NBSI ,0102 ,lanoisseforP yelseW-nosiddA .noitamotuA tnemyolpeD dna ,tseT
- [01] -dlroW etaerC ot woH :koobdnaH spOveD ehT ,elbmuH .J dna ,silliW .J ,siobeD .P ,miK .G ,sserP noitulover TI .snoitazinagro ygolohnceT ni ytiruceS dna ,ytilibaileR ,ytiligA ssalC .3008872491-879 :NBSI ,6102
- [11] ,weivrevO : (spOLM) snoitarepO gninraeL enihcaM` ,lhcsriH .S dna ,lh□K .N ,regrebzuerK .D .3202 ,978 13-668 13 .pp ,11 .lov ,ssecca EEEI " ,erutcetihcra dna ,noitinifeD

- [21] A :tfirD tpecnoC rednU gninraeL` ,gnahZ .G dna ,amaG .J ,uG .F ,gnoD .F ,uiL .A ,uL .J -6432 .pp ,21 .on ,13 .lov ,gnireenignE ataD dna egdelwonK no snoitcasnarT EEEI " ,weiveR [10.1109/TKDE.2018.2876857](https://doi.org/10.1109/TKDE.2018.2876857) :IOD .9102 ,3632
- [31] erawtfoS fo ymonoxaT A` ,civoknrC .I dna ,nossLO .H .H ,hcsOB .J ,jaR .A ,eratakawL .E .L -722 .pp " ,noitagitsevnl laciripmE nA :smetsyS gninraeL enihcaM rof segnellahC gnireenignE [10.1007/978-3-030-19034-7\\_14](https://doi.org/10.1007/978-3-030-19034-7_14) :IOD .9102 ,342
- [41] noitcudorP gniveihca :gnireenignE ytilibavresBO ,adnariM .G dna ,senoJ-gnoF .L ,srojaM .C -5446702941-879 :NBSI ,2202 ,aideM yllieR'O .ecnellecxE
- [51] .N ,lluK .M ,ollarO-zednanreH .J ,irreF .C ,odnahcO-sarertnoC .L ,demulP-zenitraM .F morF :retal sraeY ytnewT MD-PSIRC` ,hcalF .P dna ,anatniuQ-zerimaR .J .M ,ehcihaL dna egdelwonK no snoitcasnarT EEEI " ,seirotcejarT ecneicS ataD ot sessecorP gniniM ataD [10.1109/TKDE.2019.2962680](https://doi.org/10.1109/TKDE.2019.2962680) :IOD .1202 ,1603-8403 .pp ,8 .on ,33 .lov ,gnireenignE ataD
- [61] yesniKcM " ,weiveR ni edaceD flaH a dna - 2202 ni IA fo etatS ehT` ,ynapmoC & yesniKcM .eulav dna noitpoda IA no yevrus launna ,2202 ,.peR .hceT ,etutitsnI labolG
- [71] :gninraeL enihcaM gniyolpeD ni segnellahC` ,ecnerwaL .D .N dna ,amrU .G-.R ,seyelaP .A :IOD .2202 ,92-1 .pp ,6 .on ,55 .lov ,syevruS gnitupmoC MCA " ,seidutS esaC fo yevruS A [10.1145/3533378](https://doi.org/10.1145/3533378)
- [81] gninniW gnisU dna ,gnitnemelpmI ,gnipoleveD :srotacidnI ecnamrofreP yeK ,retnemraP .D .2015298111-879 :NBSI ,5102 ,yeliW .dr3 ,sIPK
- [91] ,niatnuoFaL .B dna ,uhC .M ,nolednaC .F ,noriK .D ,hednabadohK .S ,mahtobsnaR .S ,weiveR tmemeganaM naolS TIM " ,gninraeL lanoitazinagrO htiW tcapmI s'IA gnidnapxE` .GCB htiw yevrus IA launna ,0202
- [02] .sredaeL ssenisuB dna sruenerpertnE rof IA ot ediuG lacitcarP A :ssenisuB rof IA ,esoR .D .9906724841-879 :NBSI ,1202 ,sserpA
- [12] .murcS ot ediuG evitinifeD ehT :ediuG murcS ehT ,dnalrehtuS .J dna rebawhcS .K .ediug ygolodohtem murcS laiciffO ,0202 ,gro.murcS
- [22] ,gnuoy .M ,yrahduahC .V ,renbE .D ,spilliH .T ,vodyvaD .E ,nivolOG .D ,tloH .G ,yellucS .D " ,smetsyS gninraeL enihcaM ni tbeD lacinheT neddiH` ,nosinneD .D dna ,opserC .F-.J .segnellahc ecnanetniam metsys LM no repap lanimeS ,5102 ,1152-3052 .pp ,82 .lov
- [32] -namuH rof etutitsnI drofnatS " ,3202 tropeR xednI ecnegilletnI laicifitrA` ,IAH drofnatS .troper sdnert IA launna evisneherpmoC ,3202 ,.peR .hceT ,ecnegilletnI laicifitrA deretneC
- [42] ,ikazayiM .M ,ettolliovaL .A ,icitneZ .J ,nahP .D ,ervfEl .K ,canetS .C ,tnomO .N ,lieverT .M .esirpretnE eht ni gninraeL enihcaM elacS ot woH :spOLM gnicudortnI ,nnamdieH .L dna .0923802941-879 :NBSI ,secitcarp spOLM ot ediuG evisneherpmoC ,0202 ,aideM yllieR'O
- [52] .M dna ,neddaM .S ,rebuH .S ,nahtanawsiV .S ,eeL .E-.W ,maynamarbuS .H ,katraV .M .6102 ,3-1 .pp " ,tmemeganaM ledom gninraeL enihcaM rof metsyS A :BDledoM` ,airahaZ [10.1145/2939502.2939516](https://doi.org/10.1145/2939502.2939516) :IOD
- [62] " ,gniniM ataD rof ledom ssecorP dradnatS a sdrawoT :MD-PSIRC` ,ppiH .J dna htriW .R .gniniM ataD rof ssecorP dradnatS yrtsudnI-ssorC ,0002 ,.peR .hceT ,GA zneB-relmiaD
- [72] .T ,gnihcruM .S ,iksniwnoK .A ,gnoH .A .S ,isdohG .A ,nosdivaD .A ,nehC .A ,airahaZ .M gninraeL enihcaM eht gnitareleccA` ,ramuZ .C dna ,eiX .F ,ehkraP .M ,eivligO .P ,mydokyN .54-93 .pp ,8102 ,14 .lov ,4 " ,wolFLM htiw elcycefiL