

יבשה בעידן הדיגיטל – למה זה לא מצליח לנו?

פרספקטיבה טכנולוגית לדיון על "יבשה באופק"

יותם הכהן¹ ויואל יפה²

* * *

אתגרי המימוש של תפיסת 'יבשה באופק', כתוכנית הנשענת באופן עמוק על טכנולוגיות מאפשרות, נעוצים בהבדלים בין עקרונות היסוד של פיתוח בעידן הפלטפורמות שבו החומרה היא הליבה, לבין עקרונות היסוד של פיתוח בעידן הלוחמה הרשתית שבו התוכנה היא הליבה. כותבי מאמר זה סוברים כי במתודולוגיות הפיתוח העדכניות בעולם, ביניהם ניהול מוצר ופיתוח גמיש (אגילי), טמון השורש להצעיד קדימה את כוחות היבשה, תוך הפנמה של אי הוודאות הכרוכה בתהליכי פיתוח של העידן החדש.

* * *

אנו חושפים דרכים טובות יותר לפיתוח תוכנה תוך עבודה ועזרה לאחרים אלו הם ערכינו ועקרונותינו:

אנשים ויחסי גומלין על פני תהליכים וכלים;

תוכנה עובדת על פני תיעוד מפורט;

שיתוף פעולה עם הלקוחות על פני משא ומתן חוזי;

תגובה לשינויים על פני מעקב אחרי התוכנית;

כלומר, בעוד שיש ערך לפריטים בצד שמאל, **אנחנו מעריכים יותר את הפריטים בצד ימין**.

המניפסט [מנשר] לפיתוח תוכנה אגילי, 2001³

¹ יותם הכהן הוא הבעלים של חברת דואלוג לפיתוח אסטרטגי ועמית מחקר במרכז דדו.

² רב-סרן (במילי) יואל יפה הוא יועץ בכיר בחברת דואלוג לפיתוח אסטרטגי ועד לאחרונה מוביל פרויקטים טכנולוגיים באמ"ן.

³ המנשר לפיתוח תוכנה אגילי, doalog.co/agilemanifest

המאמר של ערן אורטל המופיע בגיליון זה⁴ מציג את עקרונות היסוד של תפיסת "יבשה באופק" ואת הסיבות לכך שהתפיסה לא הגיעה למיצוי מספק, למרות המאמץ הרב שהושקע במימושה והחשיבות הרבה שבכירי זרוע היבשה ייחסו לה. לא היינו מעורבים במישרים בתהליך החשיבה שתואר במאמר ולא במאמצי המימוש של העקרונות שנסקרו בו. אף על פי כן תיאור הקשיים במאמר דומה להפליא לתיאורים חוזרים ונשנים של מאמצי פיתוח טכנולוגיים שלא השיגו את יעדם. אנו סוברים כי לצד שבעת מוקדי הקושי ההקשריים שמונה אורטל, קיים עוד קושי אינהרנטי שראוי לדון בו בנפרד – בניין כוח בעידן ה־IT. הניסיון שנצבר בעשורים האחרונים ביחס לפיתוח תוכנה ומערכות טכנולוגיות מבוזרות, חיוני כדי להבין את ההקשר הרחב יותר שבתוכו נטועים חלק מן הקשיים המתוארים. מתוך דיון זה ניתן לדון בהתאמת הניסיון הטכנולוגי האזרחי לצרכים ולצורת הארגון הצבאית לטובת פיתוח תפיסת בניין כוח טכנולוגית רלוונטית יותר שתאפשר את פריצת הדרך מול אתגרי היבשה בעת הנוכחית.

על אתגרי היבשה

יחסי הגומלין של צבאות עם הטכנולוגיה האזרחית הם מורכבים מטיבם – לא פעם הצבאות נבנים בד בבד עם טכנולוגיה מתהווה, אך לא פעם הם מתקשים לעמוד בקצב הפיתוח האזרחי. עם זאת יש לציין כי במהלך המאה ה־20 הממסדים הביטחוניים הובילו לא פעם את ההתפתחויות הטכנולוגיות המשמעותיות ביותר (כך למשל בתחום הפלטפורמות האוויריות, בשימוש יישומי בלוויינות, בפיתוח אמצעי תקשורת וכינון מסד לרשת האינטרנט ובכוח החישוב שעלה בכוחו וקטן בגודלו). אולם מאז שנות ה־80 דומה כי התהפכו היוצרות, וכיום מובילה התעשייה האזרחית והשוק החופשי את חזית הטכנולוגיה, במיוחד במחשוב, במידע ואף בנגיעה בתחום הפלטפורמות (רחפנים מחד וטילים לשיגור מטען לחלל מאידך).

על רקע זה השינויים המואצים בזירה הטכנולוגית, מאז שנות ה־90 מנסים צבאות היבשה בעולם לאמץ את היכולות הדיגיטליות ולרתום אותן לצורך שינוי עמוק בצורת הלחימה. אחת הדוגמאות המרכזיות והמוכרות לניסיונות ממין זה היא תוכנית FCS (Future Combat System) האמריקאית שיצאה לפועל בראשית שנות ה־2000. תוכנית זו, שבמידה רבה הייתה הנגזרת של צבא היבשה של ארה"ב לתפיסת הלחימה מבוססת הרשת (NCW) ושל ההצלחה לתרגם אותה למערכת אווירית מבצעית מוצלחת, הייתה כישלון מפואר ודוגמה לקושי האינהרנטי להעביר את צבא היבשה לעולם הדיגיטלי. לכישלון האמריקאי היו אבות רבים:

⁴ ערן אורטל, "מבולבלים!! גם אנחנו! פרספקטיבה היסטורית לדיון על "יבשה באופק", בין הקטבים, גיליון 16-17, (צה"ל: מרכז דדו, יולי 2018).

ניהוליים, פרויקטאליים, אמל"חים ועוד, אבל הוא בעיקר הדגים את האתגרים המהותיים ביכולת להפוך את צבא היבשה לדיגיטלי.⁵

הצלחת חיל האוויר הישראלי לתרגם את היכולות הרשתיות והדיגיטליות ('פריסקופ') המתפתחות להישג מבצעי משמעותי כבר בשלב מוקדם יחסית בשנות ה־80, במסגרת מבצע 'ערצב 19', מעלה עוד יותר את השאלה למה היבשה לא מצליחה, ומה אפשר לעשות בנידון? אולם ההשוואה לחיל האוויר, שנערכת תדיר ולרוב תוך הדגשת הסטנדרט הניהולי של חיל האוויר והקושי הניהולי של היבשה, מחייבת חשיבה ביקורתית. אם נחזור לרגעי ההצלחה של חיל האוויר ב־1982 ונעמיד מולם את אתגרי היבשה, נוכל לחדד את ההבדלים:

- **פשטות (יחסית) של התמונה האווירית למול מורכבות של היבשה** – התמונה האווירית מורכבת במקרי קיצון ממאות פריטים, וניתן לשלוט עליה ממרכז שליטה אחד, ואילו התמונה היבשתית מורכבת בכל רגע נתון מעשרות אלפי גורמים שונים שלרובם אין חתימה דיגיטלית;
- **תווך אלקטרוני מלא מול סביבה שהיא פיזיקלית במהותה** – כבר עשרות שנים שלחמת האוויר מתקיימת הרחק מעבר לקו הראייה כך שהיא מתווכת באופן מוחלט באמצעים אלקטרוניים; הטווחים והמורכבות של היבשה מובילים לכך שבעתיד הנראה לעין המעטפת הדיגיטלית היא רק גורם מסייע שאינו מחליף את הלחימה הפיזית;
- **בהירות המשימה באוויר מאפשרת הקמה של מערכות מידע ייעודיות** התפורות באופן הדוק לתהליך מבצעי – הבהירות הזו אינה מתקיימת בצבא היבשה שהחיכוך עצמו הוא חלק מהותי מאופן ברור המשימה שלו.

נציין בקצרה שהמעבר של חיל האוויר וגם של חיל הים,⁶ הדומה לו במאפייני התקשורת והשו"ב, להתמודדות עם משימות אסימטריות הקשורות בתווך היבשתי (עליונות ואש במתאר לבנון או עזה) מדגימות את האתגר העצום שעיימו מתמודדת היבשה. מכל הידוע לנו לא קיים כיום קונספט המסוגל לספק מענה הרמטי ביבשה כפי שאלו קיימים באזורי האוויר והים. מורכבות זו היא גורם בסיסי ההופך את מערכות הלחימה המתוארות ב"יבשה באופק" למחויבות מצד אחד, אך גם למורכבות מאוד לאפיון ולפיתוח מצד שני, כפי שנראה להלן.

⁵ להרחבה ראו ד"ר גדעון עקביה "FCS – החלום ושברו", מערכות 458, (דצמבר 2014): 22-29.
⁶ ראו מאמר של מפקד חיל הים אלוף אלי שרביט, "מגדלור באופק", בין הקטבים 14.

בין פיתוח תוכנה לפיתוח פלטפורמה

האתגרים של תפיסת "יבשה באופק" כתוכנית הנשענת באופן עמוק על טכנולוגיות מאפשרות, נעוצים בעומקם בשוני המהותי הקיים בין פיתוח טכנולוגי 'קלאסי' הממוקד בפלטפורמת הלחימה (רק"ם, מטוס, כתמ"ם, ספינה וכדומה) או אפילו כלי נשק (רוס"ר, טיל), לבין פיתוח מערכת תוכנה. כלומר תשתית המאפשרת חיבור טכני של נקודות קצה לטובת יצירת מערכת טכנולוגית לוחמת ואחודה.

פיתוח בעידן הפלטפורמות – קווים לדמותו

כדי להבין את עומק ההבדל, ראוי להקדיש מחשבה לניסוח רעיון פיתוח האמל"ח בעידן הפלטפורמות. במהלך המחצית השנייה של המאה ה-20 הגיע המערב לעליונות ברורה, ביחס למקבילות מן המזרח, כמעט בכל פלטפורמת לחימה מרכזית – הטנק, המטוס, ספינת הטילים, הצוללת ועוד. זו הייתה עליונות שהושגה מתוך מצב של נקודת זינוק דומה שהחלה בסוף מלחמת העולם השנייה, כאשר הפלטפורמות מן המערב והמזרח היו כמעט זהות ביכולות המובנות שלהן.



תמונה 1: טנק AMX-13 בתקופת ההמתנה לפני מלחמת ששת הימים

(מאת: לשכת העיתונות הממשלתית)⁷

⁷ [https://he.wikipedia.org/wiki/AMX-13#/media/File:Flickr_-_Government_Press_Office_\(GPO\)_-_AMX_Tank.jpg](https://he.wikipedia.org/wiki/AMX-13#/media/File:Flickr_-_Government_Press_Office_(GPO)_-_AMX_Tank.jpg)

במלחמת ששת הימים ניצחו טנקי AMX-13 והשרמן הישראליים את M-60 הירדני T-55 המצרי, לא בזכות איכותם ועליונותם הטכנולוגית, אלא בזכות מקצועיות טקטית, אימון, מוטיבציה, איכות לוחמים ופוי"ש. במונחים של ימינו ניתן לראות בהיבטים אלו את הרשת המאגברת ההופכת פלטפורמות נחותות למערכת לוחמת בעלת עליונות מובהקת. אולם בנקודה מסוימת בזמן הפכה הטכנולוגיה של הפלטפורמה להיות גורם 'משנה משחק'. היבט זה קיבל עם השנים חשיבות רבה כל כך עד שהביטוי "יתרון איכותי" הפך להיות מזוהה באופן בלעדי עם יתרון טכנולוגי ולא עם יתרון אנושי.

במידה רבה השינוי המתואר הוא ביטוי מושלם להצלחת תפיסת פיתוח האמל"ח של המערב. המאמץ המתמשך של תעשיות הנשק במערב הגיע לקו פרשת מים סביב שנות ה-70 של המאה ה-20 עם עליית המרכזיות של החימוש המדויק. הגילום המובהק ביותר של עליונות זו בא לידי ביטוי במהלך מלחמת המפרץ הראשונה שהייתה ביטוי חד לפער בין הפלטפורמות המובילות מן המערב על אלה מן המזרח וגם ליכולת של המערב לתרגם את היכולות הללו לכדי תפיסה מבצעית שלמה. כבר בתיאור זה ניבטים כמה מעקרונות היסוד של הפיתוח בעידן הפלטפורמות:

- **פלטפורמות הן גנריות** – העליונות שתוארה היא עליונות של פלטפורמה מול פלטפורמה – טנק מן המערב מול טנק מן המזרח, מטוס מול מטוס, וכדומה; הגנריות מאפשרת להשוות בין הפלטפורמות ולהציב את ההשוואה כיעד פיתוח ראוי;
- **מדדי הפיתוח הם אוניברסליים** – והם מבוטאים בהיבטים פיזיקליים ברורים – חוסן השריון, טווח הפגיעה ואחוזי הדיוק, טווחי תנועה, מהירות וכדומה; גם היבט זה הוא בר השוואה במהותו; העליונות האוניברסלית של המערכות מאפשרת למכור אותן as is למדינות אחרות ללא הכרח לבצע התאמות עמוקות בתצורה;
- **יעדי הפיתוח קשורים באופן ישיר בצורך מבצעי טקטי** – הפיתוח נדרש לאפשר ללוחמים מעטפת ביצועים משופרת לפי הצורך המבצעי כפי שהוא מובן באותה העת, ולאור יכולות היריב המתהוות (שהן גם קונקרטיות ואוניברסליות במהותן);
- **עליונות היא נתון** – לאור האוניברסליות המתוארת ניתן להבין כי יש פלטפורמות שיש להן עדיפות איכותית ברורה וחד משמעית, ולפי תפיסה זו העליונות של הגוף הלוחם נובעת בראש ובראשונה מיתרון טכנולוגי;
- **התגברות על המגבלות הפיזיקליות הנתונות** – המתחים הפיזיקליים המובנים בין היעדים השונים (מיגון מול מהירות וטווח; כוח אש מול משקל וכדומה) חייבו את גורמי הפיתוח לחקור ולהמציא לאורך זמן מבנים אופטימליים הדוחקים לקצה את מעטפת הביצועים

האפשרות או הדוחפים לשיפור המעטפת הפיזיקלית (חומרים מרוכבים, מנועים יעילים יותר);

- **אמל"ח בקצה הפיתוח הטכנולוגי** – העליונות של המערב נבעה ממו"פ בסיסי שהוליד טכנולוגיות חדשות לגמרי – טכנולוגיות שלא היה להן כל קיום או שימוש במרחב האזרחי; היישום של טכנולוגיות אלו שימש להשגת פער של עשורים בין מערכות הפיתוח במערב ובמזרח;

- **גופי פיתוח סגורים** – הפיתוח התקיים בתעשיות ביטחוניות סגורות, אשר כל אחת מהן פיתחה "קופסה סגורה" מבחינה טכנולוגית גם אם לא את הפלטפורמה כולה (למשל, תותח לטנק, מערכת בקרת אש, טיל, מכ"ם ועוד);

- **זמני הפיתוח ארוכים** – כנגזרת של ההיבטים האמורים היה ברור שזמני הפיתוח עד להגעה לשלבי הצטיידות הם ארוכים מאוד (במיעוט חמש שנים ולעתים גם תמשעשרה שנים ויותר);

- **זמני השימוש ארוכים עוד יותר** – ההצטיידות בפלטפורמה היא התחייבות דרמטית משום שהיא אמורה ללוות את הצבא למשך מספר עשורים; פלטפורמות צבאיות מובילות מעורבות בפעילות מבצעית ארבעה עשורים ואף יותר מכך;

- **פיתוח מפלטפורמה לפלטפורמה ולאחר מכן מבלוק לבלוק** – ההיסטוריה של הפיתוח מראה כי בתקופה הראשונה נעשה פיתוח מפלטפורמה לפלטפורמה (למשל, F-15 כפיתוח עוקב ל-F-4 – פאנטום), ואילו בשלב מסוים ההתקדמות המרכזית חלה מבלוק לבלוק; שינוי זה נובע מהבגרות של הרכיבים והתצורה המכניים ומן השינויים הדרסטיים שחלו באותן שנים בתחום המחשוב; בעוד שמעבר מפלטפורמה לפלטפורמה ארך שנות דור, המעבר מבלוק לבלוק (למשל, טנק מרכבה סימן 1-4), היה קצר יותר, בהרבה, וארך שנים ספורות;
- **הפיתוח הוא אתגר יקר מאוד ולכן רק תעשיות מדינתיות יכולות לשאת בו חלק** – הגורם המוביל של הפיתוח בכללותו הן תעשיות הביטחון הנסמכות באופן ישיר או ישיר למחצה על המדינה כמממנת.

כל האמור היווה בסיס לתפיסת הפיתוח ובניין הכוח התואמת בעידן הפלטפורמות:

- **אפיון ותכנון מפורט לפני פיתוח** – רק באמצעות אפיון מדויק ומפורט מאוד בשלבי היציאה לדרך ניתן להביא את מערכות הפיתוח ליעילות הנדרשת לטובת צמצום עלויות הפיתוח המשמעותיות;

- **תפיסת פיתוח לינארית (Waterfall)** – תפיסת הפיתוח משלב האפיון, דרך החלוקה לתת מערכות, לפיתוח אבטיפוס ותצוגות תכלית ועד לייצור נרחב היא לינארית במהותה, ואם התהליך נדרש לחזור לשלב קודם – הרי זה בשל כשל עמוק בתהליך שעלותו גבוהה;
- **אתה תגיד את ה"מה" ואני אגיד את ה"איך"** – גורמי השטח והמפקדים היו מעורבים באופן ישיר ועמוק בפיתוח ונדרשו להבנה בסיסית בלבד של ההיבטים הטכנולוגיים, והמהנדסים נדרשו להבנה בסיסית של הצורך המבצעי לצורך מיצוי באופן אופטימלי; ההיבט המובהק ביותר של הפרדה זו התבטא בהקמת גופי פיתוח חוץ צבאיים (התעשיות הביטחוניות) במטרה לשכלל את היכולות ההנדסיות-אזרחיות ולרתום כוחות משמעותיים מהחברה התעשייתית;
- **האינטגרציה בין הרכיבים היא חלק מהתכנון** – כל פלטפורמה מורכבת בהכרח מרכיבים שונים; האינטגרציה בין רכיבים אלו היא בהכרח חלק מן התכנון הראשוני המאפשר לפתח כל אחד מן הרכיבים בנפרד בשלב מאוחר יותר.

מערכות הפיתוח הטכנולוגיות, כמו גם תו"ל בניין הכוח הצבאי, פותחו והוקמו בעידן של פיתוח הפלטפורמות, ועל כן הן מבוססות באופן ישיר על הנחות יסוד מעידן זה. אולם תפיסת הפלטפורמות כאבן היסוד של הפיתוח הביטחוני הגיעה למיצוי בגלל אי-הסימטריה מחד ובגלל ההשמדה ההדדית מבוססת חימוש מונחה מדויק מאידך. לכן נדרשת לוחמה מבוססת רשת כדי לאפשר לפלטפורמות לשוב לעליונות המוחלטת שזכו לה בעולם שבו אנו נדרשים ללחימה סימולטניות בקצב מהיר מאי פעם.

פיתוח בעידן ה-ICT – למה זה צריך להיראות אחרת?

כפי שמיטיב לציין אורטל המהפכות התעשייתיות אינן מבטלות את קודמותיהן, וגם במעבר לפיתוח בעידן ה-IT יש חשיבות רבה להמשך פיתוח בתחום הפלטפורמות. אלא שרוב הפיתוח הנדרש בהיבטי "יבשה באופק" אינו נוגע לפיתוח פלטפורמות כאלו ואחרות (בשונה מה-FCS), אלא לכינון מערכת לחימה שלמה, מבוססת ICT. אנו נוקטים בראשי התיבות ICT (Information and Communication Technology) כמבטא האתגר המרכזי של הפיתוח בעידן הנוכחי – איסוף, ניתוח, ארגון מידע והנגשתו באופן רלוונטי בתוך הרשת. על כן ראוי להבין את הבדלי העומק בין פיתוח של פלטפורמות לבין פיתוח מערכות בעידן החדש. הבנות היסוד המגולמות בפיתוח הטכנולוגי ב"יבשה באופק" המבקש לממש את רעיונות הלוחמ"ם והאינטרנט המבצעי, תפיסת האש החדשה וכן רעיונות חופת את"ר היא חתירה לקונקרטיזציה של הרעיונות הכלליים (עדיין) של המהפכה התעשייתית הרביעית ובכלל זאת

רעיון ה-IoT – "האינטרנט של הדברים". כל ההיבטים הללו אינם עוסקים בשיפור הפלטפורמות אלא בשכבה נוספת, רשתית, שתשנה מהותית את תצורת הלחימה, גם ללא החלפה בהכרח של הפלטפורמות הקיימות (לפחות לא בשלבים הראשונים). עוד לפני שנכנס לדיון בסעיפי הדברים, יש כאן בהכרח אתגר שהוא חדש ואחר מליבת הפיתוח הביטחוני עד כה, וברור כי יש לגשת אליו באופן אחר.



תמונה 2: פיתוח נשקים במעבדה בבריטניה.

(מאת: Royal Navy)⁸

אם מפרקים את האתגר המתואר למרכיביו הבסיסיים, תוך הבנת ההישגים שכבר מומשו בעידן הפיתוח הקודם (למשל בתחום הסנסורים או בתחום החימוש מונחה מדויק), נראה כי אפשר להגדיר אותו דרך שני אתגרים שונים שפתרון הכרחי לטובת התקדמות במימוש התפיסות:

- **אתגר הרישות** – היכולת לפרוס אינטרנט מבצעי בפס רחב עבור גופים שהם דינמיים במהותם (תמרון) היא אתגר טכנולוגיפיזיקלי שיש שטוענים שאינו ניתן לפתרון מלא בטכנולוגיות הקיימות כיום; אנו נתייחס להלן למרכיב זה באופן מצומצם;
- **אתגר המערכת** – גם אם תהיה הצלחה בפריסת הרשת התומכת בהספקים הרצויים, היישומיות של הרשת מחייבת פיתוח ברמת התוכנה המאפשר את החיבור בין המרכיבים

⁸<https://www.royalnavy.mod.uk/careers/roles-and-specialisations/services/surface-fleet/engineering-technician-weapon-engineering>

השונים בזמן אמת בצורה יעילה ואפקטיבית; להבנתנו התחום של פיתוח אמל"ח מבוסס תוכנה הוא נושא תפיסתי עיקרי שבו נעסוק באופן נרחב בהמשך הדברים.

המעבר מפיתוח פלטפורמה שבמסגרתו החומרה היא ליבת הפיתוח, ואילו התוכנה היא היבט מלווה לפיתוח מערכת רשתית, הופך את התוכנה לרכיב פיתוח עיקרי המאפשר, המשלים והממצה את התקשורת הרשתית, שהיא בגדר חומרה עבור המערכת. הצבאות חווים את המעבר מפיתוח מבוסס חומרה לפיתוח מבוסס תוכנה באיחור של עשור וחצי ממגוון סיבות שהראשית שבהן היא שעדיין לא ניתן לנצל מלחמות עם תוכנה בלבד (על אף 'עידן הסב"ר').

אם מונים את הייחודיות של פיתוח מערכות ICT בסדר הגדול הנדון לשימוש צבאי, יש להתייחס הן להיבטים הגנריים הנובעים מן המעבר לתצורה מבוססת תוכנה והן למאפיינים הייחודיים של מערכת לחימה.

ההיבטים הנובעים מן המעבר לפיתוח תוכנה

היעדר מגבלות פיזיקליות, אפשר לפתח כמעט הכל – המערכות הנחוצות לתהליכים ב'חופת את"ר' נמצאות הרבה מתחת לתקרת הזכוכית החומרית והפיזיקלית של יכולות המחשוב היום.

מגוון רחב וזמין הרבה יותר של טכנולוגיות, תפיסות עיצוביות וממשק משתמש – נסו לדמיין מערכת מידע המקבלת מידע מסוגים שונים של סנסורים, מעבדת אותו ומתיכה אותו לתוצר המאפשר קבלת החלטה או אף סגירת מעגל אוטומטית בזמן אמת. כמות הטכנולוגיות שבהן ניתן להשתמש לטובת מימוש כל מרכיב טכנולוגי בתוכנה זו, הוא עצום וההבדלים בין הטכנולוגיות השונות אינם בהירים בשלבי היציאה לדרך (אם קיימים בכלל). מנגד שינויים לא גדולים בתפיסת העיצוב וממשק המשתמש עשויים לייצר הבדלים דרמטיים בשימושיות של תוכנות שונות.

היעדר החשיבות של המדדים האוניברסליים – המדדים האוניברסליים בתחום התוכנה (מהירות חישוב, כמות פריטים שניתן לקבל ולעבד בו זמנית ועוד) אינם רלוונטיים לאפקטיביות של תוכנה. קשה מאוד להשוות בין שתי תוכנות שונות העושות פעולה דומה.

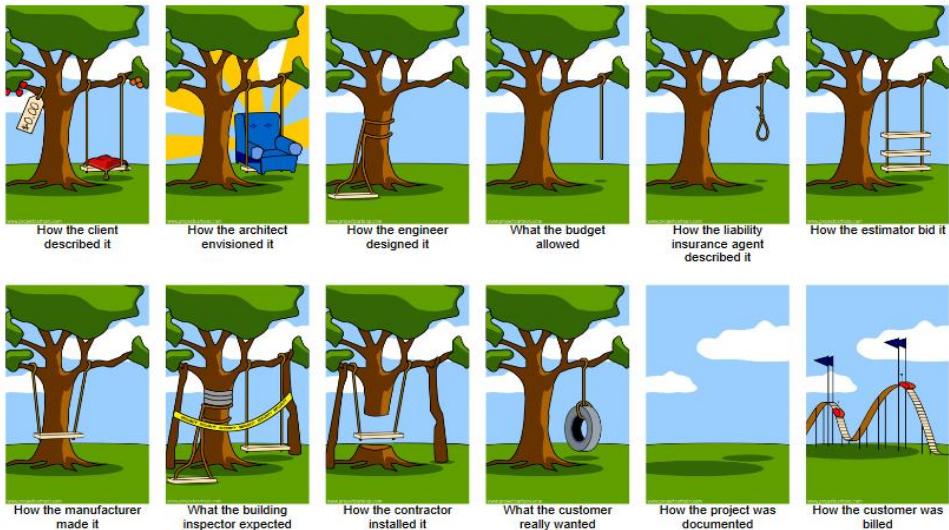
הפער בין האפשרויות האינסופיות והקוגניציה המוגבלת – השחרור מהמגבלות הפיזיקליות מחדד את מגבלות הקוגניציה האנושית. אנו מסוגלים לפתח כמעט הכל, אך מתקשים להגות ולחשוב על הכל. היבט זה בא לידי ביטוי בדיסוננס המגולם בתהליכי פיתוח רבים הנקרעים בין אפיונים מרחיקי לכת שהם בגדר פנטזיה שאין לה כיסוי טכנולוגי במסגרת משאבי הפיתוח הקיימים או שאינם עונים על צורך ממשי בשטח, לבין משקולות "ריאליסטיות" המבטאות קוצר רואי המשקע את המערכות בבעיות ההווה.

צורך מבצעי מעורפל לצד מורכבות קונספטואלית – לא תמיד ניתן להגדיר צורך מבצעי מדויק למערכת מידע, ופעמים רחוקות עוד יותר האפקט של מערכת מידע מסוימת מובן בתחילת בנייתה.

אתגר האפיון בפרויקטי תוכנה – פרויקטי תוכנה רבים נעים בין הרצון והדרישה לאפיון מלא ומפורט (עד כדי אלפי עמודים) לבין העובדה שאפיון כזה אינו בהכרח תנאי להצלחה ולעיתים הוא אף מוביל לכישלון הפרויקט.⁹ האתגר הגדול ביותר בפיתוח מערכת תוכנה בסדר הגודל הנידון הוא הבנה משותפת, ומסמך מפורט וחתום איננו הכלי היעיל ביותר לבניית ההבנה המשותפת הזאת.

The Project Construction Cycle - The Tree Swing

Create your own cartoon at www.projectcartoon.com



איור 1: תרשים פופולארי אודות הפער בין אפיון והתוצר הסופי¹⁰

⁹ כפי שאמר פעם אחד הקמנ"רים: "לצערנו, שוב המתכנתים עשו בדיוק את מה שביקשנו מהם".
¹⁰ משותף מתוך projectcartoon.com/cartoon/46980 ברישיון CC BY 3.0.

האינטגרציה בין הרכיבים מתקיימת בסוף התהליך – קשה מאוד לתכנן אינטגרציה בין חלקי תוכנה שונים. בהיעדר היערכות נכונה, שאינה רק עוד שלב בלוח הזמנים של הפרויקט, שלב האינטגרציה הופך להיות ארוך, קשה ומסורבל, לעתים ברמה המסכנת את הצלחת הפרויקט כולו. כאשר מדובר באינטגרציה עם מערכות ישנות ('legacy') אזי רמת הסיבוכיות עולה עוד יותר. כך, באחת המערכות הזרועיות המרכזיות – המקושרת לאינסוף מערכות וממשקים – אורך מחזור הפיתוח שנה: שלושה חודשי פיתוח ותשעה חודשי אינטגרציה. אומנם ישנן דרכים שונות לפשט את התהליך הזה – תקנים, ארכיטקטורה, ניהול וכדומה – אך עדיין מדובר באתגר מרכזי.

גמדים על כתפי גמדים – כל פיתוח תוכנה מתבסס על תשתיות תוכנה קודמות. גוף פיתוח שאינו מתעדכן בפיתוח התדיר, דן עצמו לקיבעון טכנולוגי¹¹ וגורם לעלייה משמעותית בעלויות הפיתוח בשל היעדר מיצוי של רכיבים קיימים.

זמני פיתוח קצרים ושינויים טכנולוגיים תכופים במהלך הפרויקט – קבועי הזמן בפיתוח תוכנה קצרים יותר בהרבה ביחס לפרויקט טכנולוגי של חומרה. פיתוח של מערכת בינונית ממוצעת (כולל שלב אפיון) אורך כשנתיים-שלוש. אולם ברוב התחומים אנו יכולים להניח מראש שבמהלך התקופה הזאת תהייה התפתחות טכנולוגית משמעותית העשויה להשפיע על האפיון ואולי אף על הבנה מחודשת של הדרישה המבצעית.

זמני שימוש קצרים – בעידן הנוכחי¹² נבנות רוב מערכות המידע לצורך שימוש של שנים בודדות ומתוך הנחה כי הטכנולוגיה והשינוי בצרכים יגרמו לצורך בהחלפתן. **עיסוק גובר בתחום של חוויית משתמש וממשק משתמש (UI/UX)** – בעוד שתחום זה נתפס בעולם הפלטפורמה כסוגיה משנית, הרי שבעולם התוכנה הוא משמעותי יותר בהרבה, ופעמים הוא ההבדל בין מערכת טובה למערכת גרועה.

הפרדוקס של עלויות פיתוח התוכנה – תוכנה היא מוצר זול מאוד ויקר מאוד בו זמנית:

- **זול** – תוכנה מחייבת מעט מאוד תשתיות פיזיות וחומרי גלם יקרים, ניתן לשכפל אותה ללא עלות, ומחירי החומרה התומכת נמצאים בירידה מתמדת;
- **יקר** – פיתוח תוכנה מחייב מיצוי של מספר רב של שנות אדם מיומן, ואם נעשה פיתוח מאפס, מדובר בתהליכים מורכבים מאוד המאופיינים באי ודאות גבוהה הבאה לידי ביטוי ברכיב התמחור.

¹¹ ראו בהקשר זה את הביקורת של מייקל אייזנברג ב'מניפסט החומס' על התמכרות הצבא, הממשל וההייטק הישראלי לתשתיות מייקרוסופט כבר ב־2010. doalog.co/humus.

¹² בשנות השמונים, התשעים ותחילת שנות ה־2000 נבנו מערכות לטווח שימוש של כ־10-15 שנה, גם לאור השינוי האיטי דאז בטכנולוגיה. שרידים למערכות אלו נמצאים בשימוש מבצעי עד היום.

מוצרי תוכנה אף פעם לא גמורים – עולם התוכנה מבוסס על כך שלא חייבים לחכות לגרסה סגורה ומהודקת, וניתן לעלות עם גרסה חלקית ולהוסיף לה או לשנות אחר כך עוד יכולות ותכולות.

הפער – היעדר תפיסת פיתוח תוכנה רשתית ביבשה

אם מתבוננים על הפער המתואר, מובן למה פיתוח המערכת הוא אתגר מורכב כל כך. גופי בניין הכוח ניצבים שוב בחזית הפיתוח הטכנולוגי. אולם עתה ליבת האתגר אינה במו"פ הטכנולוגי הבסיסי אלא בפיתוח הקונספט וביישומו. הפיתוח נדרש להיבחן לא ביכולות המדף שלו, אלא בשימושיות והאפקטיביות המערכתית שהוא מאפשר. כדי לעשות כן נדרשים גופי הפיתוח לגבש תפיסת פיתוח המותאמת לעידן הלחימה הרביעי. אנו סוברים כי התבוננות תפיסתית רצינית במתודולוגיות המובילות מתחום פיתוח התוכנה בשנים האחרונות והתאמתן לסביבה הביטחון היא תשתית רצינית לקידום תפיסה מסוג זה, ועל כן נפרוט בשורות שלהלן מספר רעיונות מרכזיים מתחומים אלו.

פער זה מובהק עוד יותר בשאלות של פיתוח לצבא היבשה. הסיבות לכך נובעות משני מרכיבים עיקריים: הצורך עמוק יותר – אתגור העליונות של צבא היבשה הוא ברור וחרף באופן הפוגע בליבת התפקוד של הזרוע (התמרון), ואילו הפער בדיגיטציה עמוק יותר גם הוא כפועל יוצא ממורכבות סביבת הלחימה בצד הכחול והאדום. למעשה האתגר של הטמעת מערכת מסוג זה בסביבת היבשה מייצר הכרח לפתח צורת פיתוח אחרת, שתיתן מענה למורכבות מבלי לעצור את הפיתוח.

מ' Waterfall ל' Agile

אחד השינויים הגדולים שעבר עולם התוכנה בעשור האחרון, הוא ההבנה ששיטת הניהול הפרויקטאלית הקלאסית (Waterfall) מובילה לא אחת לכשלים עמוקים בתהליך הפיתוח כאשר מחילים אותה על פרויקטי תוכנה.

הפרוייקטים שהביאו לידי ביטוי את בעיית שיטת המפלים באופן חריף ביותר הם פרויקטים ממשלתיים, שהושקע בהם מראש סכום כסף נכבד. כך, למשל, סקירה של פרויקטי תוכנה גדולים שניהלה ממשלת ישראל בשנות ה-2000 הציגה אחוזי כישלון גבוהים מאוד, חוסר הבנה והלימה בין הדרישות לפיתוח, עיכובים בלוי"ז וחריגה משמעותית מהעלות המוערכת¹³. ניסיון נוסף לתכנן את המערכת של כל המערכות בפרוייקט המחשוב של הביטוח הלאומי הגיע גם

¹³ ראו דו"ח ממשלתי בנושא משנת 2010 : doalog.co/bigproject

הוא לאחרונה לידי סגירה וסיום כושל¹⁴, וגם הצבא חווה מספר לא זעום של כשלונות מן הסוג הזה¹⁵.

התיאור שנערך בתחילת המאמר אודות מאפייני הפיתוח בעידן התוכנה מבהיר מדוע תכנון מפלי הוא גישה בעייתית ביותר כאשר עוסקים בפרוייקטי תוכנה:

1. **רכיב אי הוודאות** – אנחנו לא באמת יודעים בתחילת הדרך את כלל מאפייני המערכת, ואם אנחנו מנסים לאפיינם באופן חלוט, אנו נוטים ליפול לדיסוננס של תכנון יתר המקדם פיצורים שוליים על פני שימושיות ואפקטיביות;
2. פיתוח תוכנה הוא מאמץ יקר מאוד, אך פיתוח תוכנה בכיוון שגוי הוא מאמץ יקר עוד יותר;
3. **קשה מאוד לדמיין ולתכנן את האינטגרציה בתחילת התהליך**, וכשמגיעים לעסוק בה, זה כבר מאוחר מדי (כי הרכיבים השונים נבנו באופן המקשה מאוד על החיבור);
4. **אנחנו לא באמת יודעים כיצד משתמשים וארגונים יעבדו עם המערכת** – קשה מאוד להבין את המיקודים השונים לפני שיש לנו נתוני אמת של שימוש במערכת (ולא הוכחה או תצוגת תכלית למעטפת ביצועים פיזיקלית), ונתונים אלו מצויים רק כאשר יש מוצר ממשי המגיע למשתמשי הקצה.

במידה רבה הבנות אלו הן הביטוי המיידי של הפער בין פיתוח חומרה לפיתוח תוכנה. על רקע מבוכות אלו החלו להתפתח החל מראשית שנות ה-90, וביתר שאת בעשורים האחרונים, שיטות ניהול אחרות וגמישות יותר¹⁶ המתבקשות להתמודד עם קריסת מתודולוגיית הפרוייקט המסורתית. גישות אלו שזכו לתיאור הכולל והמוכר "Agile Software Development", מבקשות להציב מענה ראוי לבעיות האינהרנטיות שבגישת Waterfall. להלן מספר נדבכים יסודיים מתוך עקרונות הפיתוח האגילי, אשר רובם עוסקים במנגנונים למניעה של פערים שיהיה קשה עד בלתי אפשרי לגשר עליהם בשלבים האלה:

- **מיקוד ב-Minimal Viable Product (MVP)** מוצר חיוני מינימלי – מטרתו להנגיש את הפיתוח בשלב מוקדם ככל הניתן למשתמשים לטובת פיתוח בסביבת השימוש ומיצוי פידבק משתמשים שממנו ימשך הפיתוח לשלבים הבאים, במקום פיתוח מלא בטרם

¹⁴ "המחדל: כך שרף ביטוח לאומי 600 מיליון שקל ו-7 שנות עבודה" "דה מרקר" 14/11/17
<https://www.themarker.com/technation/1.4593533>

¹⁵ לדוגמה: "חיל האוויר שוקל להחליף את נס TSG ואלביט בהובלת פרויקט ענק, לאחר שלא עמד ביעדיו" 'אנשים ומחשבים', 21/2/13, 110998, pc.co.il/it-news/110998

¹⁶ Peter Varhol, "To agility and beyond: The history and legacy of agile development", *TechBeacon* doalog.co/agilehistory

- ההנגשה ללקוחות; להבנתנו יש חשיבות רבה להמשגת MVP גם לנוכח מאמצי פיתוח נרחבים מן הסוג הנדרש לחופת את"ר;
- **אינטגרציה מתמשכת (Continuous Integration)** – במהלך כל שלבי הפיתוח ולא רק בשלבי הסיום;
- **פיתוח בסבבים מהירים (איטרציות) ועדכוני גרסה תכופים** – באופן זה ניתן לצמצם את המורכבות הרבה של תהליכי הפיתוח הארוכים ולהביא את מערכות הפיתוח ליעילות רבה יותר; את הפיתוח מבלוק מבלוק מחליף פיתוח מבוסס איטרציות קצרות (ספרינט);
- **צוותי פיתוח אחודים** – בעוד שהפיתוח בגישת המפלים התבסס לרוב על צוותים המבוססים על טכנולוגיות (בסיס נתונים, צד לקוח וכדומה) או רכיבים נפרדים, פרויקטים אגיליים מתבססים על צוותים אחודים המקדמים את המוצר או נדבך שלם במוצר בכל איטרציה;
- **פידבק משתמשים ישיר** – לנוכח הקשיים הקוגניטיביים לצפות מראש את התנהגות המשתמשים הגישה מקדמת שיח ישיר ומתמשך עם הלקוח על גבי המערכת הנבנית לצורך הבנת הצורך המבצעי האמיתי.

פיתוח אגילי כניהול מערכה עמוקה

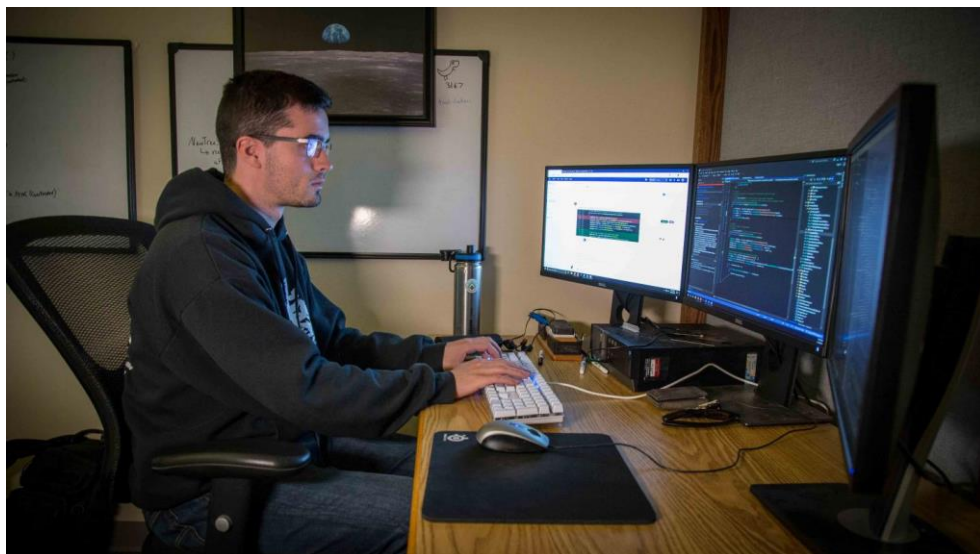
התיאור של עקרונות היסוד של הפיתוח האגילי לא אמור להיות זר למערכות צבאיות. בכל האמור להפעלת הכוח ולתוכניות בתחום הפעלת הכוח אנו מניחים אינהרנטית כי רב הנסתר על הנגלה, וכי הערפל ואי הוודאות הם מאפייני היסוד של שדה הקרב. מול מציאות זו פיתוח הצבאות יכולות לצמצום מרכיבי אי הוודאות בזמן אמת, הן ברמה הטקטית (תוך שימוש במודיעין וסיוור) והן ברמה המערכתית (תוך מימוש חיכוך לטובת למידה ופיתוח מהלומה). צורת חשיבה זו באה לידי ביטוי באופן השלם ביותר בעולם היבשה, וכמעט אין לה ביטוי בלחימת האוויר כיום.¹⁷

דווקא בתחום הפיתוח עדיין סבורים הצבאות כי ניתן להגיע ליציבות ולתכנון תבוני אופטימלי. הגם שחשיבה זו קידמה מאוד את הטכנולוגיות הצבאיות בעידן הפלטפורמות, היא הגיעה לסוף דרכה מבחינת היכולת לקדם פיתוח של מערכות שו"ב מתקדמות מן הדור הרביעי עבור צורכי היבשה.

כמו ביטויים רבים מעולם החומרה המסרבים להשתחרר בעולם פיתוח התוכנה, המושג המרכזי שיש לדון בו ולשקול מחדש את השימוש בו הם גופי האמל"ח. הביטוי אמל"ח מכיל בתוכו את כל מכלול ההנחות הקשור בפיתוח פלטפורמות, והוא מציב את מפקדי המחלקות

¹⁷ דב תמרי, "הכוח האווירי לאן", מערכות 437, (יוני 2011): 11. maarachot.idf.il/PDF/FILES/5/112925.pdf

בפוזיציה של מנהלי הפרויקט (במקרה הטוב) או כמוטבים משפיעים (במקרה הרווח יותר). החיבור ההדוק הנדרש בעידן התוכנה בין פיתוח ובין יישום נמנע בשל ההגדרות האלו ובשל עולמות הידע שבהם מחזיקות מחלקות האמל"ח (ידע מבצעי ומתודולוגי אך לא פיתוחי טכנולוגי).



תמונה 3: סטודנט מאוניברסיטת מונטנה במעבדת פיתוח תוכנה למערכות אמל"ח.

(מאת: ErekaAlert)¹⁸

התפיסה האמל"חית בצה"ל בנויה על הפרדה בין הגורם המבצעי הדורש ('הלקוח') לבין הגורם הכותב את מסמכי האפיון ('אמל"ח') המתווך בין הגורם המבצעי לבין הגוף המממש, לבין גוף המימוש (תעשייה או יחידה טכנולוגית צה"לית).¹⁹ לתפיסה זו יש יתרונות בעולם הפלטפורמה אך היא כושלת בפיתוח מערך תוכנה ליבתי.²⁰

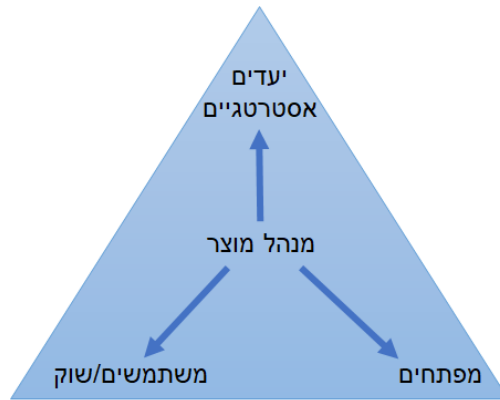
גם בעולם מערכות המידע הייתה נהוגה במשך שנים תפיסה דומה, ואת תפקיד אנשי האמל"ח מילאו מנתחי המערכות וכותבי האפיונים. אלו ישבו עם הלקוחות המבצעיים וכתבו ספרי אפיון עבים בני אלפי עמודים שנועדו להבהיר למתכנתים מה הם אמורים לעשות. גישה זו מוחלפת בשנים האחרונות בתהליכים דינמיים ואחודים המובלים על ידי מנהל מוצר.

¹⁸https://www.eurekaalert.org/pub_releases/2018-03/msu-mtl032918.php

¹⁹ בחיל האוויר יש אפילו צלע נוספת לתהליך הזה – להק ציוד. לא ניגע כאן בתפקידו ובנסיבות שהובילו להתפתחותו.
²⁰ נסים חניה, "תמורות במערכת הפיתוח והייצור הביטחונית הישראלית ומידת התאמתה לעידן הנוכחי", בין הקטבים 6, (ינואר 2016).

מחלקת מוצר כמערכת של בניין הכוח

את ההגדרה של מחלקות האמל"ח היינו מבקשים להתידך עם הרעיון של מנהל המוצר (Product). מנהל המוצר הוא גורם מאמצע קריטי בהצלחה של חברות שהצלחתן מבוססת פיתוח, והוא נדרש להימצא בתווך שבין יכולות הפיתוח הקיימות במערכת (המה והאיך – משאבים, ויכולות טכנולוגיות – מה אפשר לפתח, הערכת זמנים, היכרות עם פיתוח דומה בעבר, וטכנולוגיות קיימות בעולם) לבין היעדים האסטרטגיים של החברה (לשם מה?) ובין דרישות הלקוח והמשתמשים (עבור מי). לאלו ניתן להוסיף רגל נוספת העוסקת במאפייני סביבת השוק שאותם הוא נדרש להכיר (או במאפייני המערכת היריבה בהקשר הביטחוני).



איור 2: מערכת המתחים של מנהל המוצר. קיים דמיון רב למתחים של מנהל מערכת

האמצע הוא אירוע מתמשך שבמסגרתו מקבל המוצר את הצורה הנדרשת לו. בתהליך, מתוך הצורה המתהווה, משתנים בהכרח גם היעדים האסטרטגיים ולעיתים משתנה גם השוק עצמו ("חינוך שוק"). מתוך הבנה שהיכולות הטכנולוגיות עצמן מצויות בשינוי תדיר, ברור שמנהל המוצר הוא פונקציה קריטית להצלחה של פרויקטים בעלי עוגן פיתוח מרכזי.²¹

אנו סבורים כי מחלקות האמל"ח מחזיקות כיום ידע נרחב בצורך של המשתמשים (הצד השמאלי של המשולש בדיאגרמה) ובהיכרות סבירה עם היעדים האסטרטגיים של הגופים שאותם הם משרתים ("יבשה באופק" במקרה שלנו), אך הם חסרים מאוד בהבנה של סביבת הפיתוח, יכולותיה ואופן המיצוי של אלו. יתרה מכך, כל תהליך ייצור המוצר אינו מתקיים במחלקות האמל"ח אלא בגופי פיתוח חיצוניים משל היה מוצר סגור. מאחר והאתגר המרכזי

²¹ ראוי להשוות משולש זה למשולש שהציב ד"ר צבי לניר כתשתית לתהליכי חשיבה אסטרטגיים המחייב ברור של "בית, סביבה ואפקט רצוי", המודל המערכתי לחשיבה אסטרטגית, אתר דואלוג.

הוא פיצוח וגיבוש תפיסת הלחימה המתקיימת בהלימה על מערכת טכנולוגית תומכת, היא לא יכולה להיות מפותחת באופן חיצוני על בסיס אפיון בפרויקט "מפלי". האפיון הזה אינו קיים באמת משום שעדיין אין אנו מבינים מספיק, וכדי שנבין עלינו להתחכך באופן מהיר, תוך יישום תהליכי A/B testing, איטרציות פיתוח מהירות ופידבק משתמשים רציף וביקורת. כלומר בעוד שתפיסת הפיתוח החיצונית הייתה מספקת בעידן של פיתוח פלטפורמות, היא לא יכולה להתקיים באופן ראוי שעה שהמערכת צריכה לגבש בעצמה את המשמעויות של הלחימה בעידן הרשתי – בטח לאור כל האתגרים הספציפיים של היבשה, כפי שתוארו לעיל.

המסמך של "יבשה באופק", מפורט ככל שיהיה, לא יכול להיות יותר מאשר הצבת יעדים אסטרטגיים עבור תהליכי הפיתוח. כדי שאלו יהפכו למציאות, יש צורך במגוון רב ומורכב של תהליכים תומכים. הפיתוח עצמו דוחק בחלקו את מעטפת הטכנולוגיה המוכרת, ואילו בחלקו האחר הוא חותר למימוש נאות של טכנולוגיות קיימות בתוך המערכת הצבאית. אם הפיתוח של כל רכיב במכלול של "יבשה באופק" מגלם איזודאות מרמות שונות, הרי שאופן השימוש של הצבא ביכולות אלו הוא עלום עוד יותר, וזו השאלה המרכזית. רק חיכוך עמוק ומתמשך של השטח עם היכולות בפיתוח יוביל ליצירת מערכות רלוונטיות שישרתו את המטרות שלשמן פותחו.

את האיטרציות התכופות הנדרשות בפיתוח מערכות בסדר הגודל שעליהן דובר ב"יבשה באופק", ניתן בהחלט לכוון כמערכה המגלמת רמות אי וודאות עמוקות כמעט כמו מערכת לחימה.

מהו ה-MVP של חופת את"ר? מיהו מנהל המוצר שלו, והאם הוא יכול להיות איש 'עולם הפלטפורמות הקלאסי'? מהן האיטרציות המרכזיות בתהליך הפיתוח, ואיך אנו מייצרים את החיכוך הנדרש ללמידה? אם נענה על שאלות אלו, נוכל להתקדם בקפיצת הדרך אל האופק הדיגיטלי.

רשימת מקורות

- אורטל, ערן. "מבולבלים!! גם אנחנו!" *בין הקטבים* 16, (יוני 2018).
- אייזנברג, מייקל. "מניפסט החומוס",
<http://sixkidsandafulltimejob.blogspot.com/2010/07/hummus-manifesto-part-1.html>
- "הועדה לבדיקת פרויקטים גדולים של פיתוח תוכנה בממשלה – דו"ח מסכם" מרץ 2010,
- הטוני, יוסי. "חיל האוויר שוקל להחליף את נס TSG ואלביט בהובלת פרויקט ענק, לאחר שלא עמד ביעדיו", *אנשים ומחשבים* (21 בפברואר 2013).
- חניה, נסים. "תמורות במערכת הפיתוח והייצור הביטחונית הישראלית ומידת התאמתה לעידן הנוכחי", *בין הקטבים* 6, (ינואר 2016).
- נויכמן, רפאלה. "המחדל: כך שרף ביטוח לאומי 600 מיליון שקל ו-7 שנות עבודה" *דה מרקר*, 14/11/17.
- עקביה, גדעון. "FCS החלום ושברו". *מערכות* 458. (דצמבר 2014): 22-29.
- שרביט אלי. "מגדלור באופק", *בין הקטבים* 14.
- תמרי, דב. "הכוח האווירי לאן", *מערכות* 437, (יוני 2011): 11.
- 'ארכיטקטורה מוכוונת שירותים', *ויקיפדיה*.
- המנשר לפיתוח תוכנה זריזה בתוך 'פיתוח תוכנה זריז', *ויקיפדיה*.