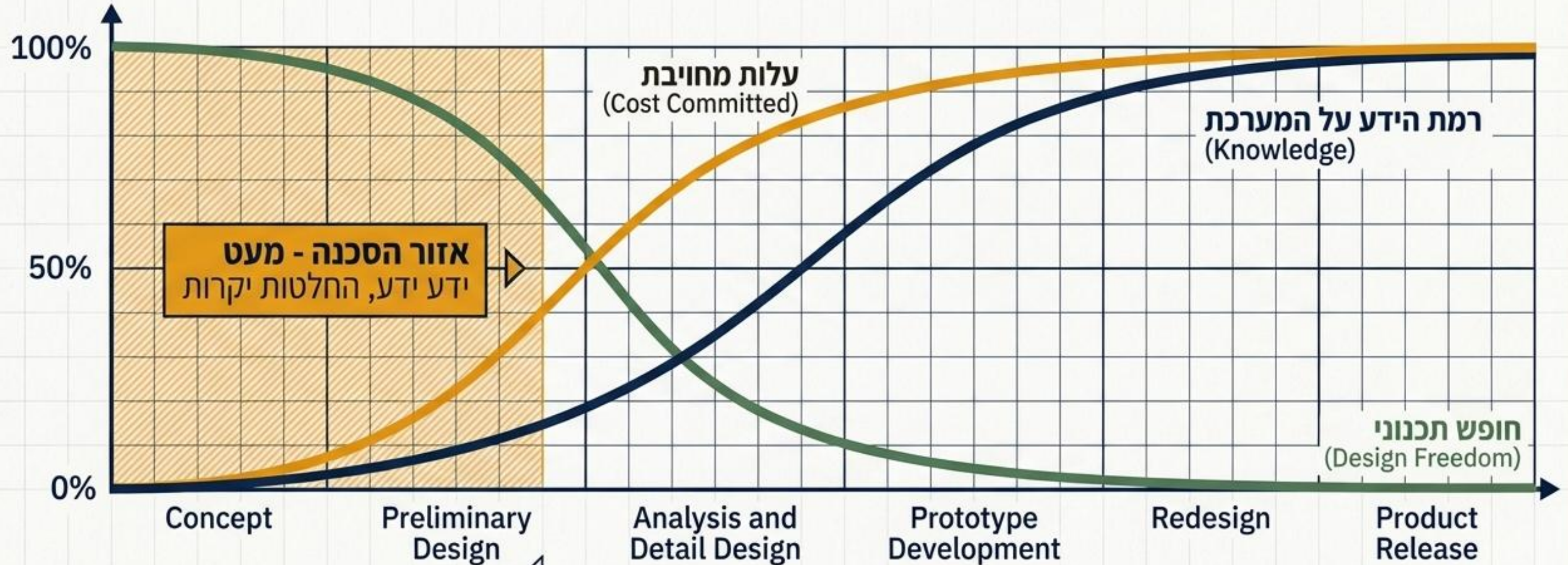


מודלים של ניהול פיתוח (SDLC)

מהנדסת מערכות למציאות משתנה: בחירת
המודל הנכון להתמודדות עם פרדוקס הידע

האתגר המרכזי: פרדוקס הידע

בפרויקטים יצירתיים, מרבית ההחלטות הקריטיות מתקבלות בשלבים המוקדמים – בדיוק כשרמת הידע שלנו היא הנמוכה ביותר.



אזור הסכנה - מעט ידע ידע, החלטות יקרות

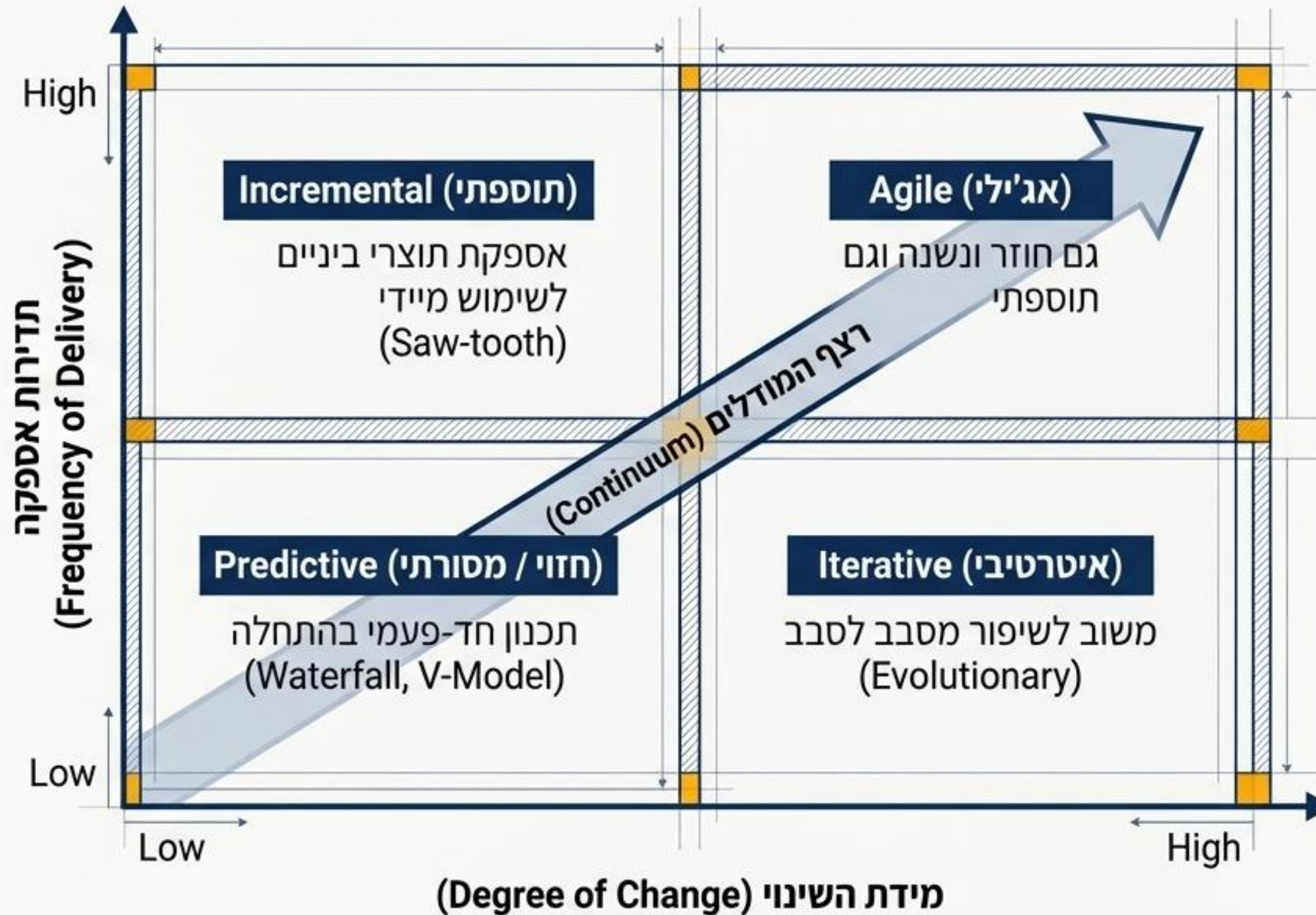
עלות מחויבת
(Cost Committed)

רמת הידע על המערכת
(Knowledge)

חופש תכנוני
(Design Freedom)

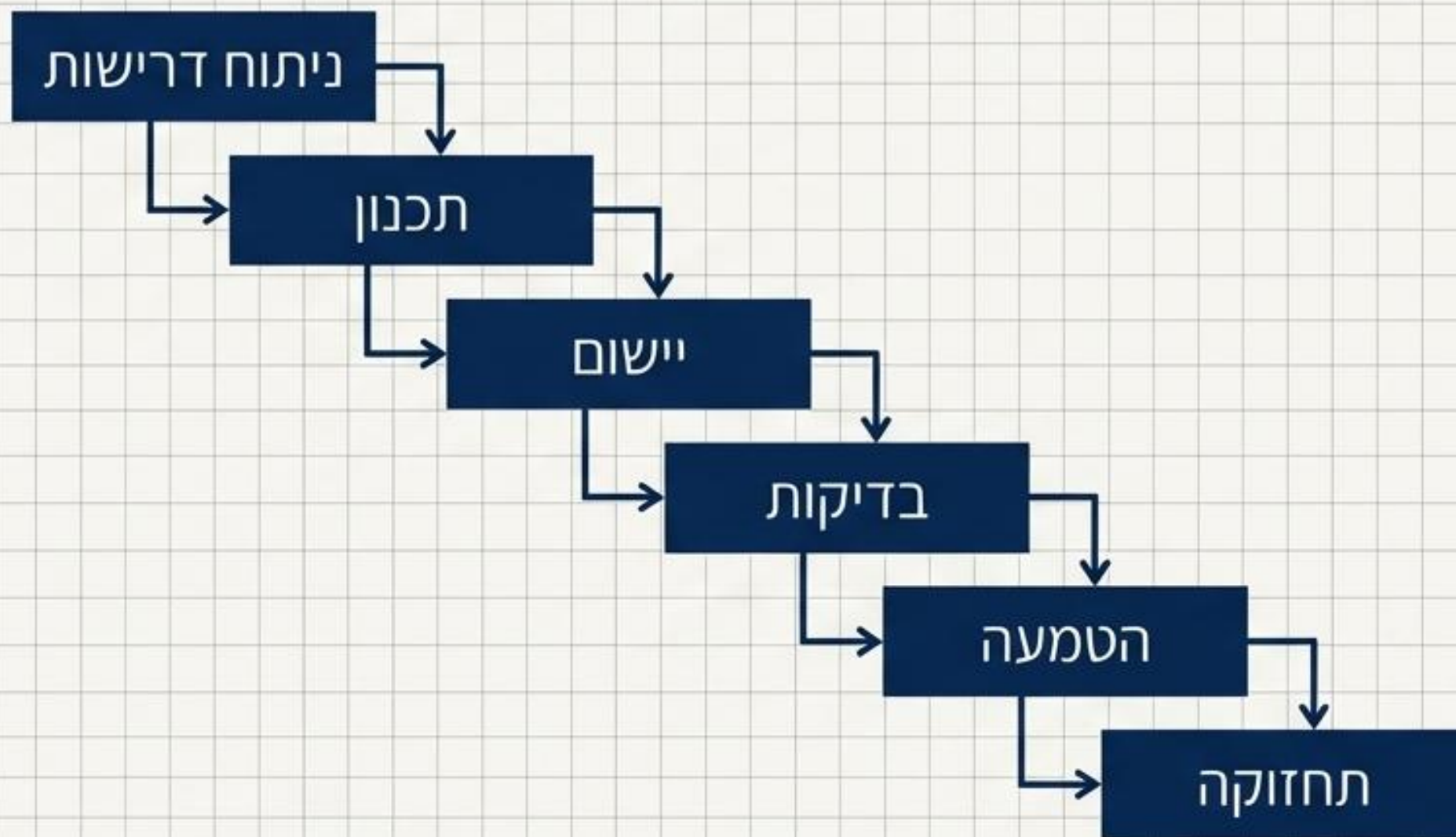
המטרה של כל מודל פיתוח היא למשוך את עקומת הידע שמאלה, ולהוריד סיכונים ואי-ודאויות בשלבים מוקדמים.

רצף הפיתוח: מחיזוי להתאמה



מודל מפל המים (Waterfall Model)

פיתוח דגם יחיד בסיום התהליך.



✓ יתרונות

- מודל פשוט ומובנה.
- פעילות אחת מסתיימת ומפעילה את הבאה אחריה.

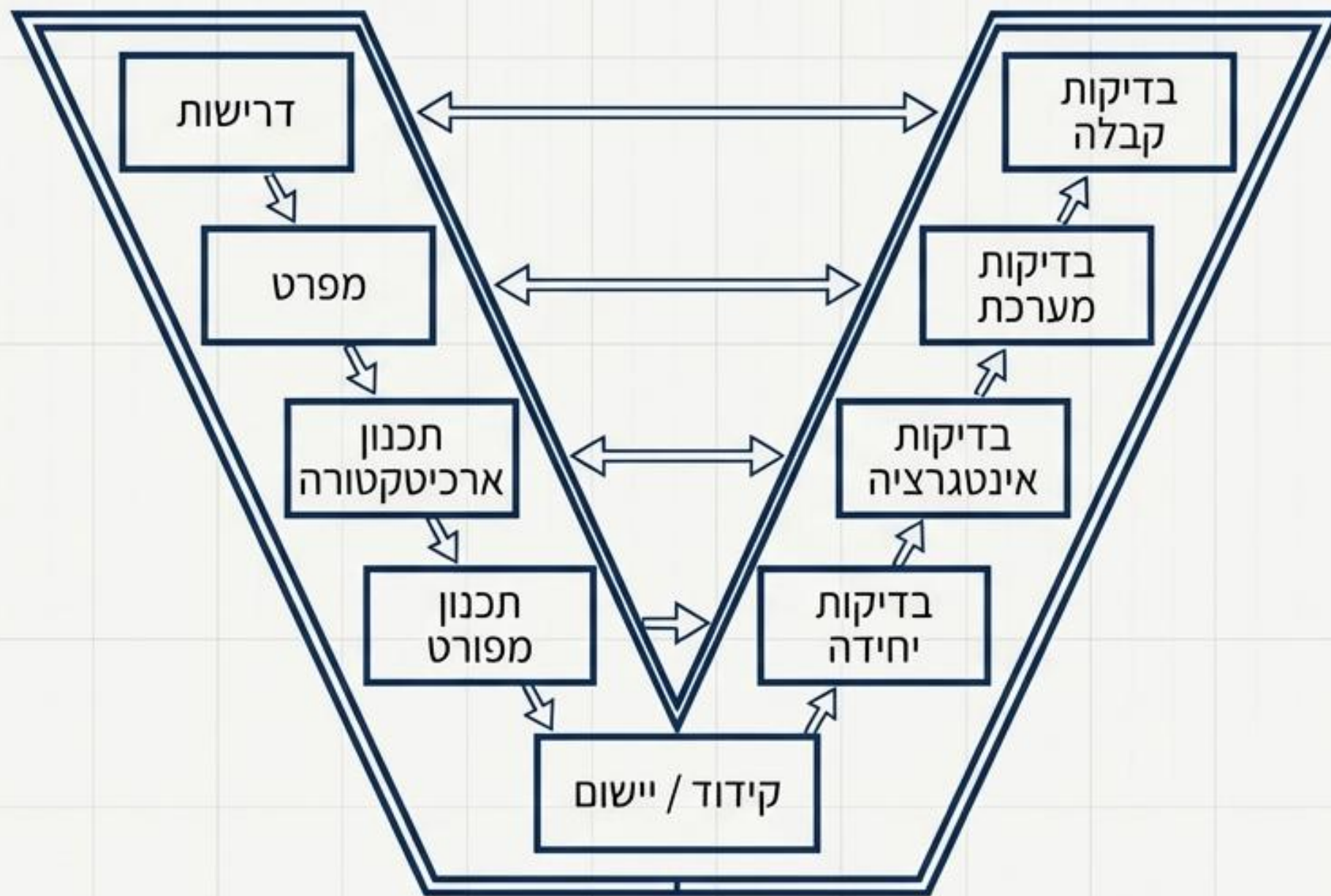
✗ חסרונות

- מודל מיושן ללא גמישות.
- המוצר מתקבל רק בסוף התהליך.

מיועד לפרויקטים קטנים, דרישות ברורות לחלוטין וסיכון נמוך.

מודל ה-V (V-Model)

מודל העבודה הטיפוסי ברפאל.



יתרונות: ✓

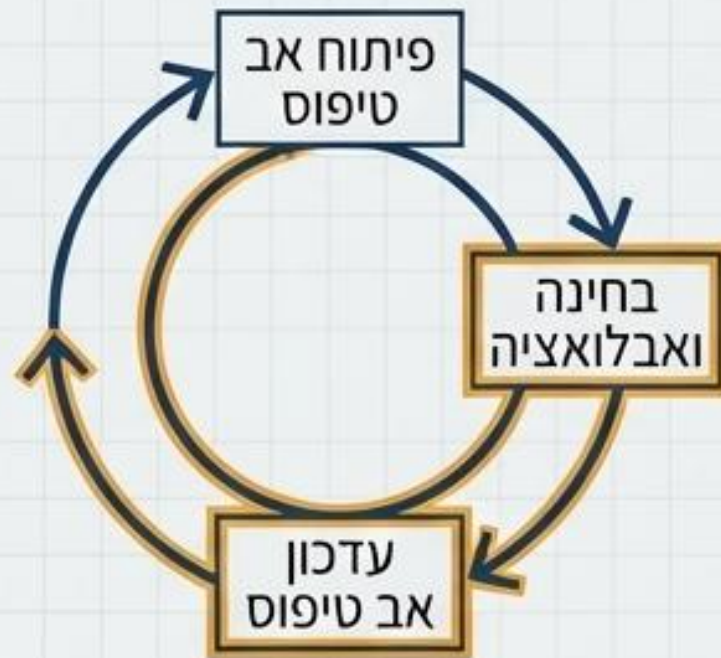
פשוט, מובנה וקל לניהול. לכל שלב פיתוח מוגדרות בדיקות מתאימות מראש.

חסרונות: ✗

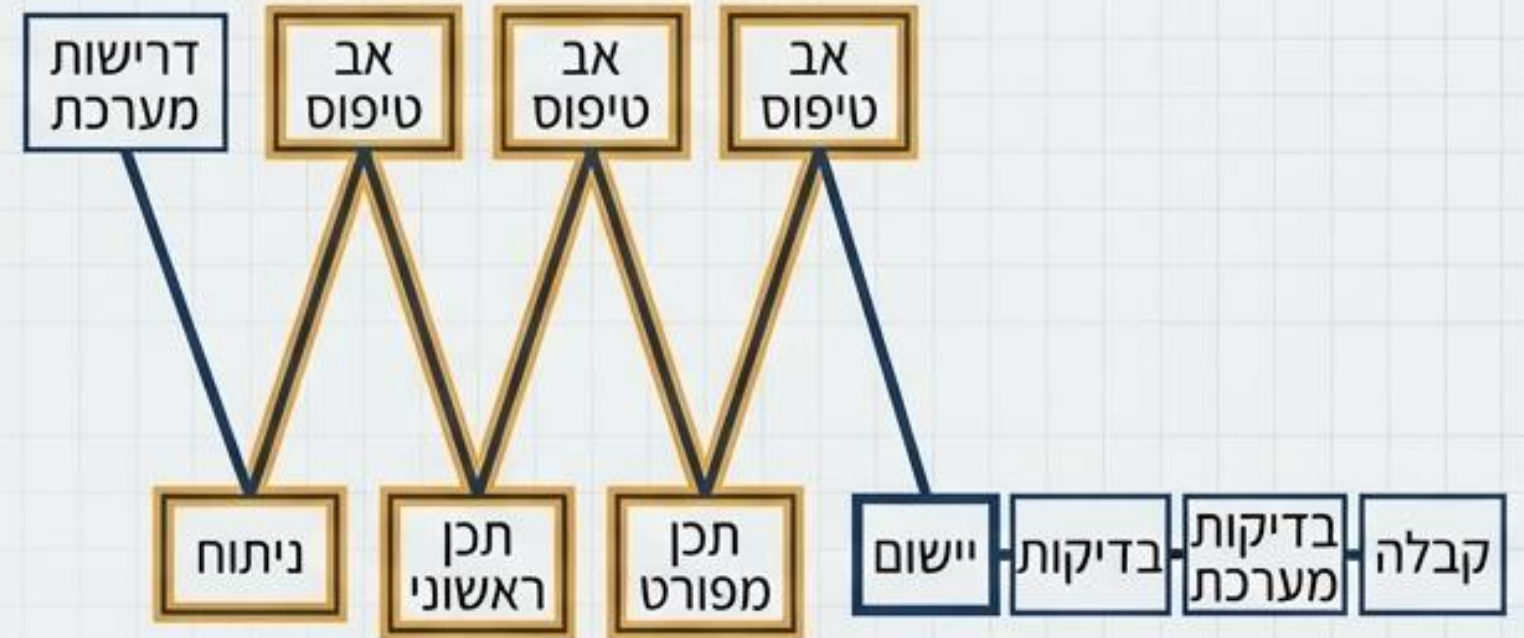
1. המערכת מתחילה לעבוד מאוחר מאוד.
2. המודל רגיש ביותר לשינויים בדרישות.
3. חשיפות קיצונית לפרדוקס הידע: קשה מאוד להעריך התקדמות אמיתית במהלך שלבי הפיתוח.

מודלים של אבות טיפוס: הורדת סיכונים מוקדמת

מודל אבולוציוני (Evolutionary)



מודל שן משור (Saw-tooth)

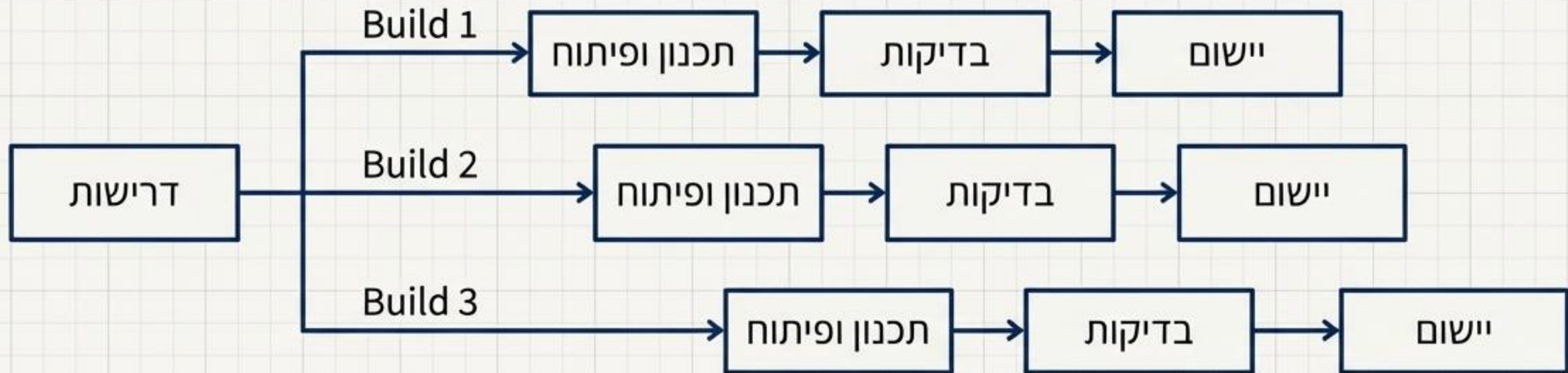


- ✓ תפקודים קריטיים נבדקים מוקדם; רגישות נמוכה לשינויי דרישות.
- ✓ דורש שילוב לקוח פעיל (או משתמש מבצעי רפאלי).
- ✗ חיסרון: תחזוקת דגמים עלולה להיחשב כבזבז.

- ✓ הורדה מוקדמת של סיכונים עיקריים נקודתיים.
- ✗ חיסרון: סכנה ליצירת מצג שווא של התקדמות מהירה אצל הלקוח/הנהלה.
- ✗ סכנה: תכן זמני עלול להפוך לקבוע.

פיתוח תוספתי (Incremental Development)

פירוק לגורמים ליציאה מהירה לשוק.



מאפיינים עיקריים:

- טכנולוגיה מפותחת ומסופקת במקביל.
- פחות רגיש לשינויי דרישות (תוצרים מתקבלים לאורך הדרך).



אזהרה ארכיטקטונית:

המודל דורש ארכיטקטורה מערכתית מודולרית מאוד כדי לאפשר התקדמות מקבילית (למשל: פיתוח מצלמת יום ומצלמת לילה במקביל). ללא מודולריות, המודל ייכשל.

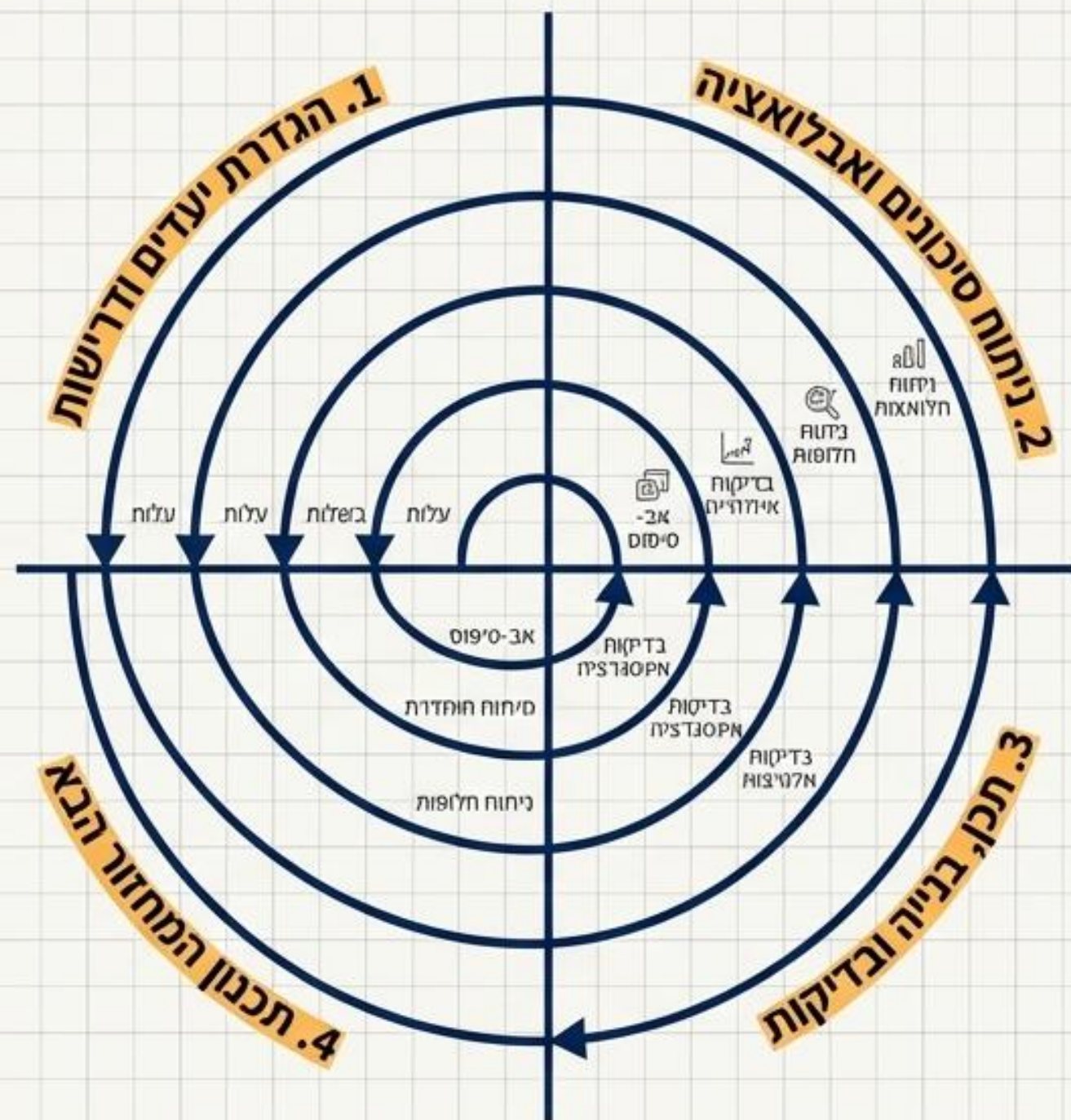
המודל הספירלי (Spiral Model)

נשק כבד לפרויקטים גדולים ועתרי סיכון.

מתי להשתמש במודל זה?

- פרויקטים גדולים ומערכות מורכבות.
- הדרישות לא סגורות לחלוטין וצפויות להשתנות.
- סביבה המאופיינת בסיכונים טכנולוגיים וניהוליים גבוהים.
- קיימת מוכנות לעלויות ניהול ותכנון בדיקות גבוהות.

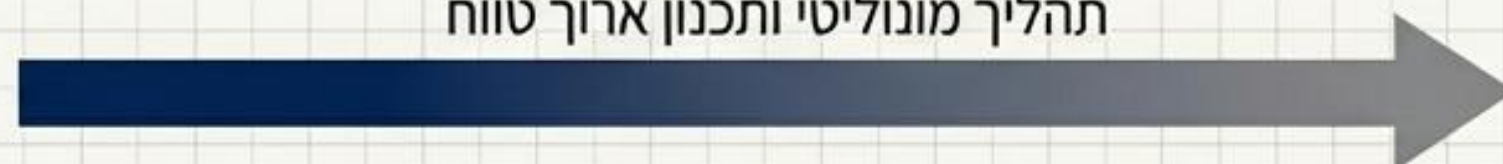
כל שלב בספירלה מפיק תוצר ברור
לאבלואציה.



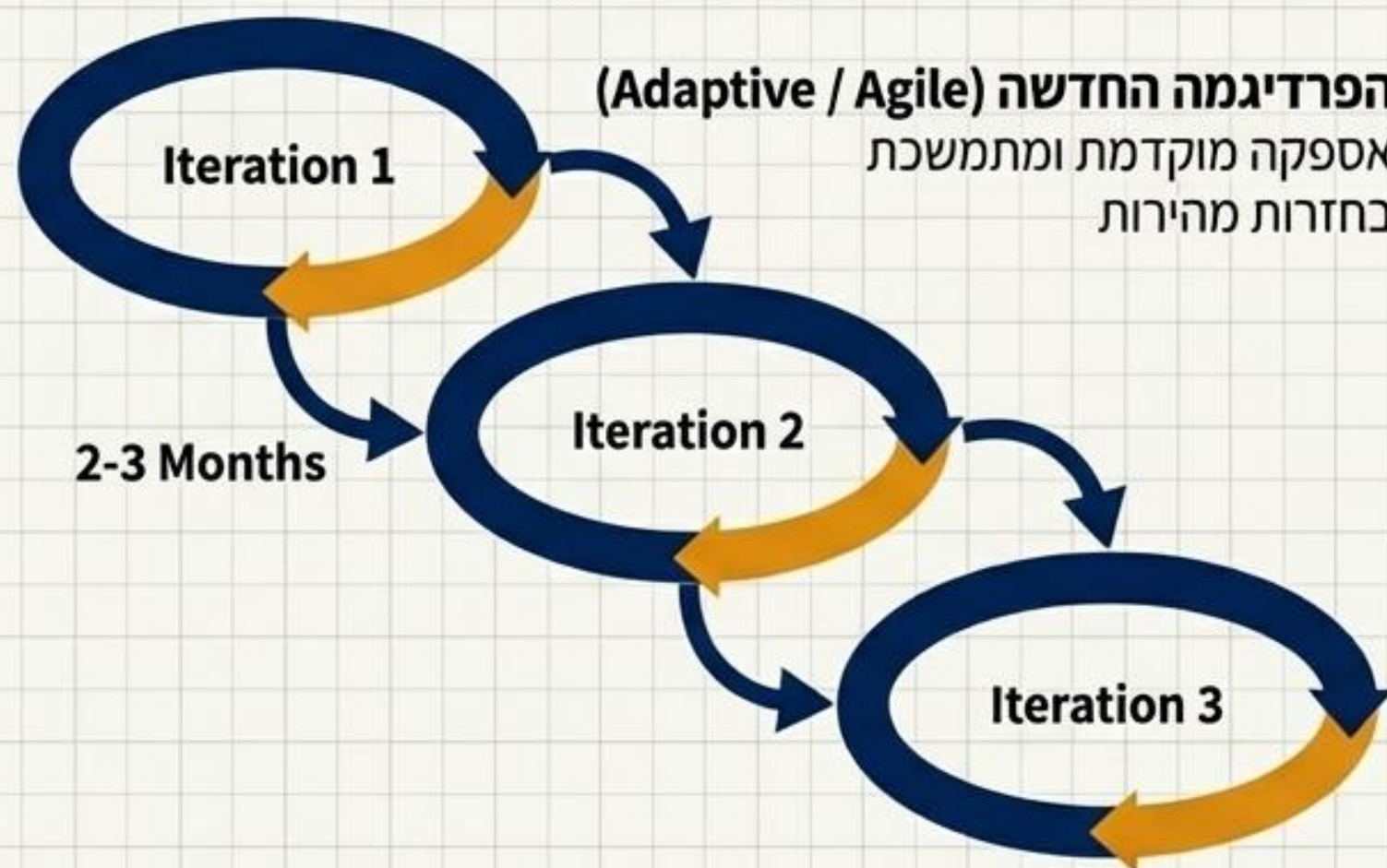
שינוי התפיסה: גישת ה-Agile

אספקה מוקדמת ומתמשכת תוך חיבוק שינויים.

הפרדיגמה הישנה (Predictive)
תהליך מונוליטי ותכנון ארוך טווח

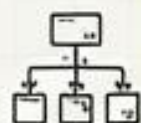


הפרדיגמה החדשה (Adaptive / Agile)
אספקה מוקדמת ומתמשכת
בחזרות מהירות



1.

חלוקת התהליך למשימות קטנות ותדירות גבוהה של אספקה (שבועיים עד חודשיים).



2.

מערכת עובדת היא המדד העיקרי והאמיתי להתקדמות.



3.

שינויים בדרישות מתקבלים בברכה - אפילו בשלבים מאוחרים.



4.

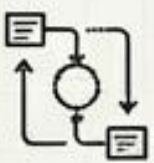
תקשורת יומיומית ואינטנסיבית פנים-אל-פנים בין המפתחים ללקוח.




האמת על אג'יל: יתרונות מול מציאות הנדסית

ההבטחה (היתרונות)

המציאות בהנדסת מערכות וחומרה



דרישות ניתנות לשינוי בקלות יחסית.



הלקוח מקבל ערך עסקי במהירות מרבית.



צוותים אוטונומיים בעלי מוטיבציה גבוהה.



מימוש קשה: קשה בהרבה למימוש בפיתוח חומרה והנדסת מערכת לעומת פיתוח תוכנה.



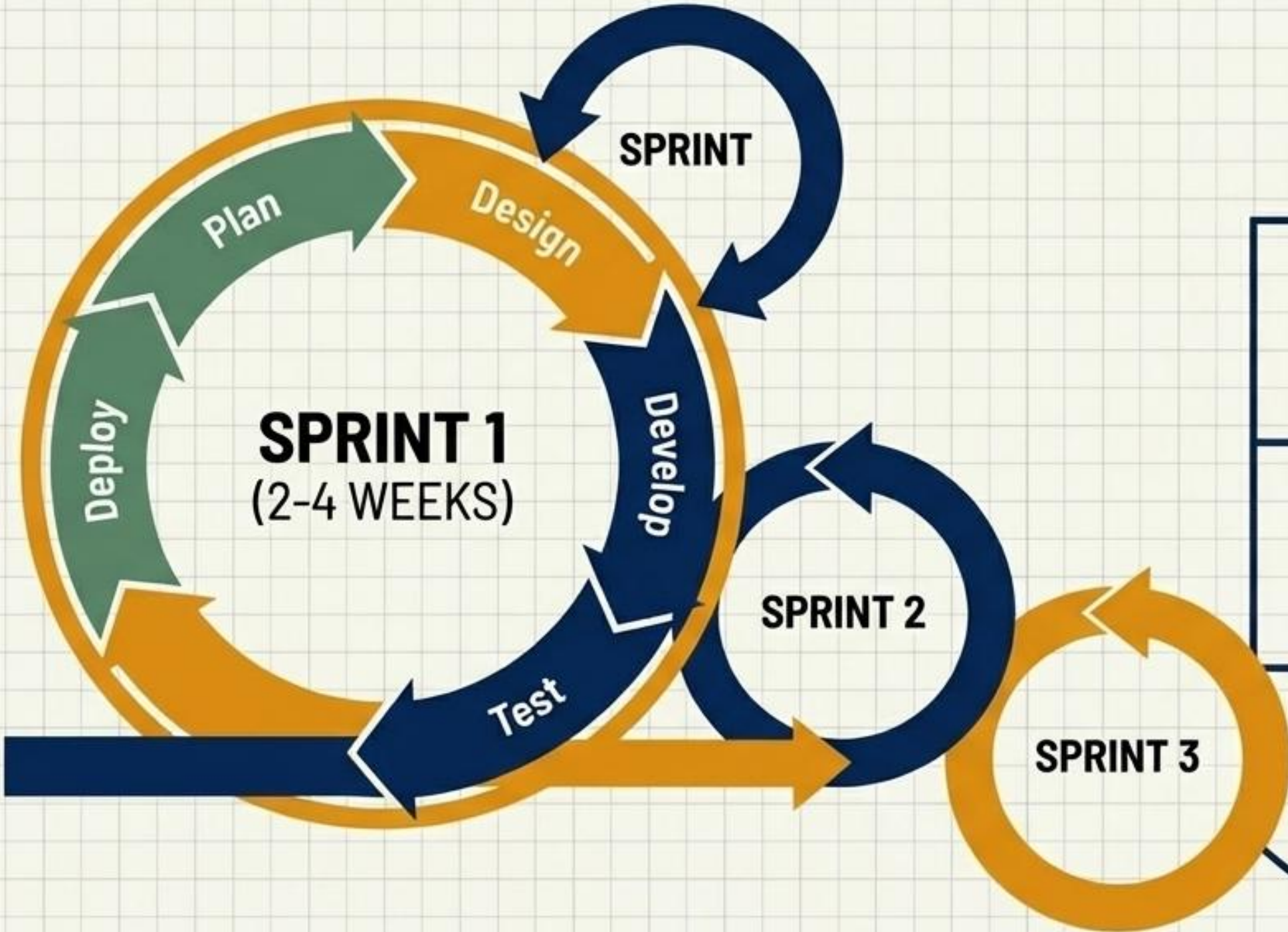
סכנת התבררות: ללא לקוח זמין וממוקד, הצוות עלול לסטות לכיוון שגוי לחלוטין ולשרוף משאבים.



תלות ארכיטקטונית: הארכיטקטורה המערכתית חייבת לאפשר פיתוח בספרינטים קצרים.

Scrum – אג'יל הלכה למעשה

מקסימום ערך עסקי בספרינטים קצובים.



Sprints: מחזורי פיתוח נעולים ובלתי משתנים של 2-4 שבועות.



The Team: צוות אורגני, ממוקד ואוטונומי של 4-9 אנשים.



The Rule: אין לבצע שום שינויים בתוך ספרינט שהוגדר. התאמות מבוצעות אך ורק בין הספרינטים.



















The Goal: בסוף כל מחזור מספקים תכולה עובדת בעלת הערך העסקי הגבוה ביותר.




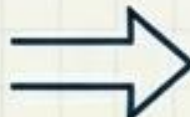










מודלים של ניהול פיתוח (SDLC)

מהנדסת מערכות למציאות משתנה: בחירת
המודל הנכון להתמודדות עם פרדוקס הידע

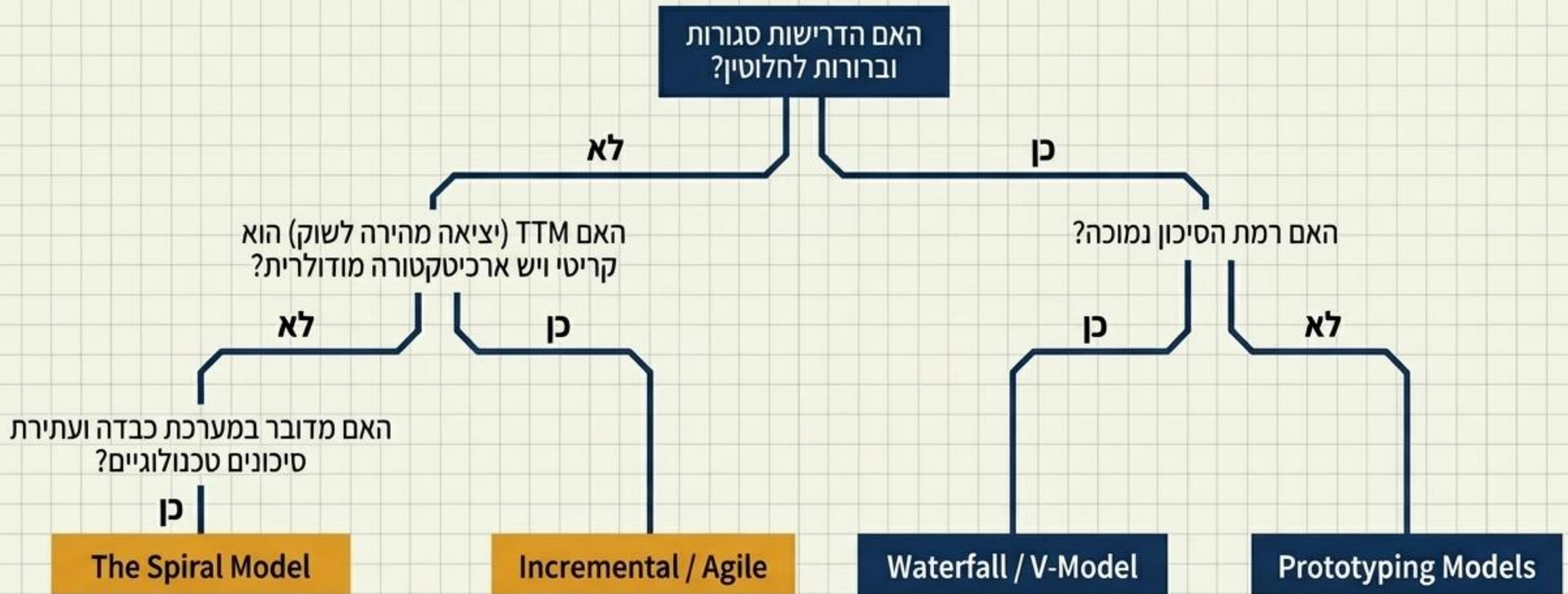
מטריצת בחירת מודל: מסורתיות ומונחות-סיכון

המודל הספירלי	מודל שן משור	מודל ה-V	מודל מפל המים	
 לא סגורות	 משתנות	 ברורות	 ברורות לחלוטין	קריאות דרישות
 גבוה מאוד	 גבוה/נקודתי	 נמוך	 נמוך מאוד	רמת סיכון נדרשת
 רציף	 אבלואציה מהירה	 מדורג לפי שלב	 בסוף הפיתוח	תדירות בדיקות
 לאחר אבלואציה מחזורית	 דגמים הולכים וגדלים	 מוצר סופי יחיד	 מוצר סופי יחיד	צורת מסירה

מטריצת בחירת מודל: אדפטיביות ומהירות

מודל אבולוציוני	פיתוח תוספתי	Agile / Scrum	
 רגישות נמוכה	 בינונית	 אפסית - מחבק שינוי	רגישות לשינויים
 בינונית	 קריטי	 מידי ורציף	מהירות הגעה לשוק (TTM)
 גבוהה	 בינונית	 קריטית יומיומית	מעורבות לקוח
 סטנדרטית	 מחייב ארכיטקטורה מקבילית	 חייבת לתמוך בספרינטים	אילוץ ארכיטקטורה

נווט הפיתוח: איך בוחרים נכון?



המציאות אינה טהורה: בפרויקטים אמיתיים משתמשים בקומבינציות. לדוגמה ספירלי ברמת הנדסת המערכת, ופיתוח Agile ברמת התוכנה בתוך האיטרציות.

סיכום: המחיר של בחירה שגויה

1. נטרלו את פרדוקס הידע

פרדוקס הידע תמיד יפעל נגדכם. בחרו מודל שמנטרל אי-ודאות מוקדם ככל האפשר בהתאם לאופי הייחודי של הפרויקט שלכם.

2. אין פתרון קסם אחד

טעות בבחירת המודל היא **הטעות היקרה ביותר** שקשה עד בלתי אפשרי לתקן תוך כדי תנועה. הכירו את החסרונות של כל מודל, לא רק את יתרונותיו.

3. תפרו את החליפה מראש

אל תאמצו מודלי פיתוח בעיוורון. שלבו, התאימו, ותכננו את מחזור החיים השלם של המערכת עוד לפני כתיבת שורת הקוד הראשונה.