

מתודולוגיות בדיקות תוכנה - חלק א

בדיקות תוכנה שיעור 1

מבוא להבטחת איכות תוכנה

הקדמה לשלב ב'

מבוא להבטחת איכות

- QA?
- פעולות לניהול האיכות



בדיקות תוכנה

- מושגים
- סוגי בדיקות
- מסמכי בדיקות



פרויקט בדיקות



היכרות



מה נלמד בשיעור?

הקדמה לשלב ב' של הקורס

אבטחת איכות תוכנה ומושגי יסוד

קופסה שחורה / לבנה / אפורה

בדיקות סטטיות / בדיקות דינמיות

סביבות עבודה

סטטיסטיקות אחרונות טוענות שגלישה של 100-200% בזמני פיתוח ובתקציב הפיתוח של מערכות תוכנה הינה גלישה כמעט מקובלת (!)

בארה"ב בלבד, גלישה זו מסתכמת ב-59 מיליארד דולר!!

בנוסף, מפתחים "מבזבזים" 34% מזמנם על תיקוני תקלות.



1. קיימת בעיה בתהליך פיתוח התוכנה
2. איכות המערכת נמוכה על אף הגלישה

הפתרון?

הבטחת איכות תוכנה – QA (Quality Assurance)

מכלול התהליכים בארגון הבאים להבטיח את איכות תפקודו ומוצריו.
במסגרת זו נכללת פעילות בקרת האיכות (QC),
אך גם נהלי העבודה ותקנים, תהליכים של סקרי תכנון,
ניהול תצורה, בקרת מסמכים, עמידה בלוחות זמנים
ומגבלות תקציב ועוד.





בקרת איכות תוכנה – QC (Quality Control)

מכלול תהליכי הבדיקה, הבאים להבטיח את התאמת תפקוד הארגון ומוצריו לתכנון.
במסגרת זו נכללים תהליך בדיקת המוצרים ותהליכי הייצור, אך גם ניהול בקרת שינויים, ניהול ובקרת מסמכים ועוד.



מהי מערכת?

אוסף של תוכניות מחשב, בשילוב עם הגדרות של מאגרי נתונים מתאימים.



מהי מערכת איכותית?

- ✓ מספקת את דרישות הלקוח
- ✓ פשוטה וזולה לתחזוק
- ✓ עומדת ביעדי זמן ותקציב
- ✓ אין בה תקלות חמורות

באג/תקלה

מונח כללי לטעות/שגיאה בקוד התוכנה,
הגורם לתוצאה בלתי צפויה או
להתנהלות לא נכונה של המערכת.

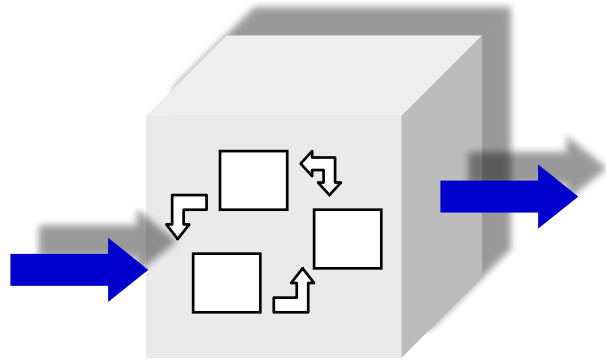


למה יש תקלות?

- חוסר שלמות אנושית
- אפיון לוקה בחסר
- בעיות תקשורת ואי הבנות
- אתגרים טכנולוגיים מסובכים



1. הבנת סוג ומבנה המערכת
2. הבנת מסמכי הדרישות והאפיון
3. הכנת תוכנית בדיקות
4. כתיבת תסריטי בדיקה מתאימים
5. הכנת סביבת העבודה ונתונים לבדיקה
6. הרצת הבדיקות
7. דיווח תקלות

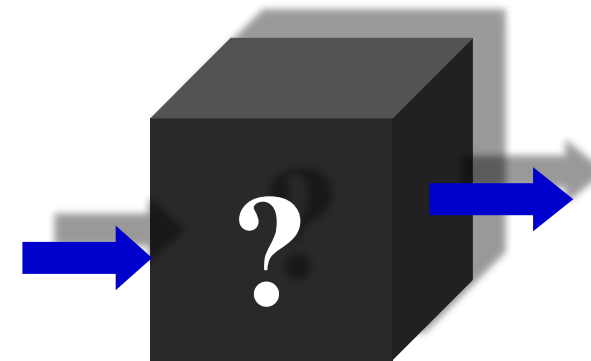


בדיקות קופסה לבנה - בדיקת איכות התכנות

- ◀ בדיקת מסלולים
- ◀ בדיקת תנאים
- ◀ בדיקת זרימת מידע
- ◀ בדיקת Loops

בדיקות קופסה שחורה - בחינת התנהגות המערכת והתאמה לדרישות

- ◀ בדיקות קלט - פלט
- ◀ בדיקות פעולה ותגובה
- ◀ בדיקות ביצועים
- ◀ בדיקות כשל והתאוששות
- ◀ בדיקת MTBF





בדיקות סטטיות:

- סקירת הקוד, מסמכי האפיון וכו'
- כשלים לדוגמא:
- חריגה מסטנדרטים בקוד
- כשלים במסמכי אפיון
- כשלים בתכנון
- חוסר כיסוי דרישות

בדיקות דינאמיות:

- הרצת התוכנה הנבדקת
- השוואת התוצאות המתקבלות לתוצאות הצפויות.

- סביבת הפיתוח (Development Environment)
- סביבת הבדיקות (Testing Environment)
- סביבת הייצור (Production Environment)

סביבת הייצור	סביבת הבדיקות	סביבת הפיתוח	
משתמשים	בודקים	מתכנתים	פעילות
תוכנת אמת נתונים "חיים"	תוכנה נבדקת נתונים לבדיקה	תוכנה בפיתוח	תכולה

אז מה למדנו?



✓ מה ההבדל בין QA ל-QC?

✓ מהי מערכת איכותית?

✓ מדוע יש תקלות?

✓ תפקיד הבודק

✓ טכניקות בדיקה

✓ סביבות עבודה

מה נלמד במפגש הבא?

✓ מודלים לפיתוח תוכנה

✓ מודל מפל המים

✓ מודל V



QA?

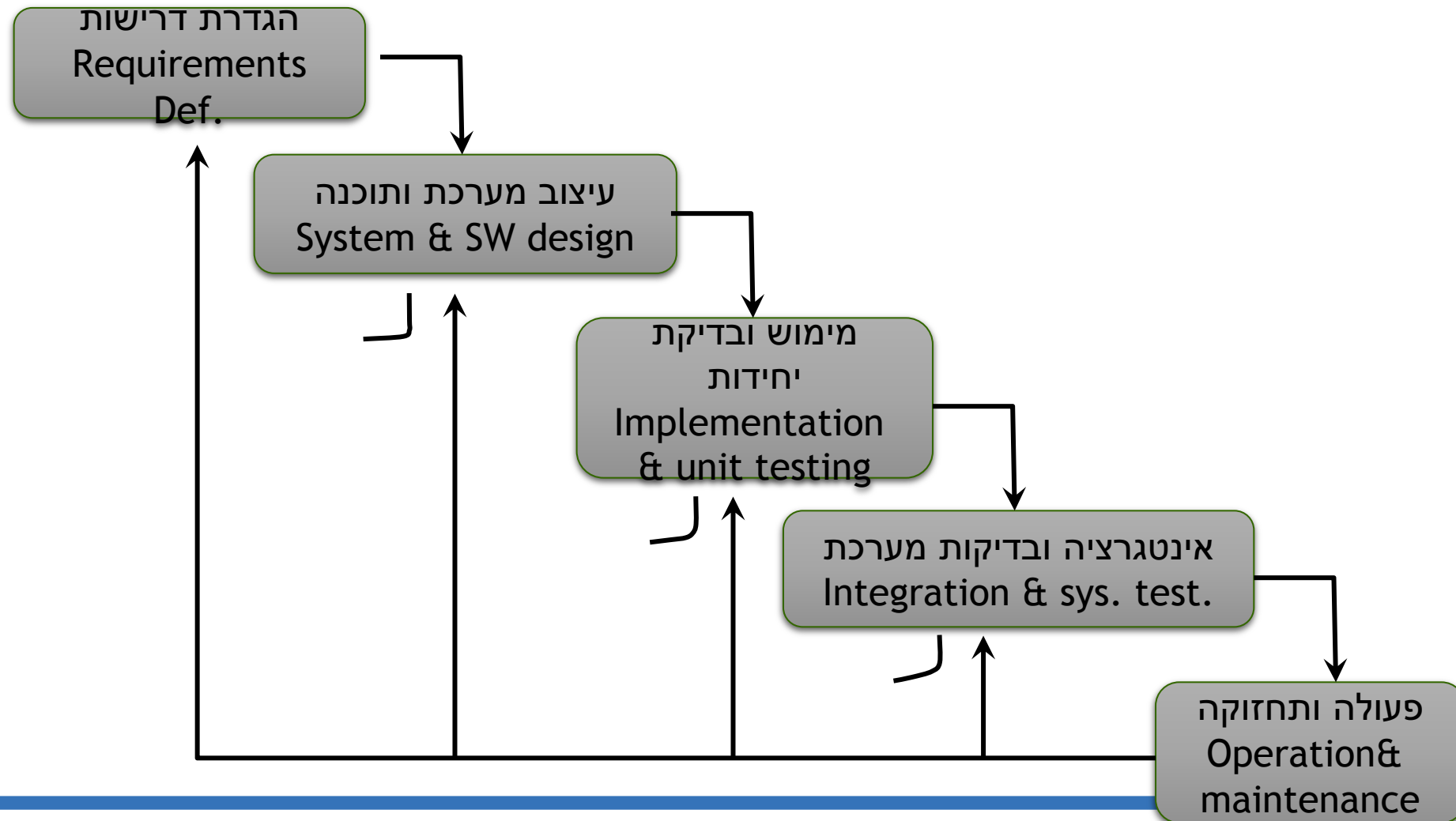
בדיקות תוכנה שיעורים 2-3

בדיקות תוכנה

מה נלמד בשיעור?

	מודלים לפיתוח תוכנה	
	מודל מפל המים (Waterfall model)	
	מודל V (V-model)	
	המודל הספירלי (Spiral model)	
	המודל הזמיש (Agile model)	
	תרגיל	

Waterfall model





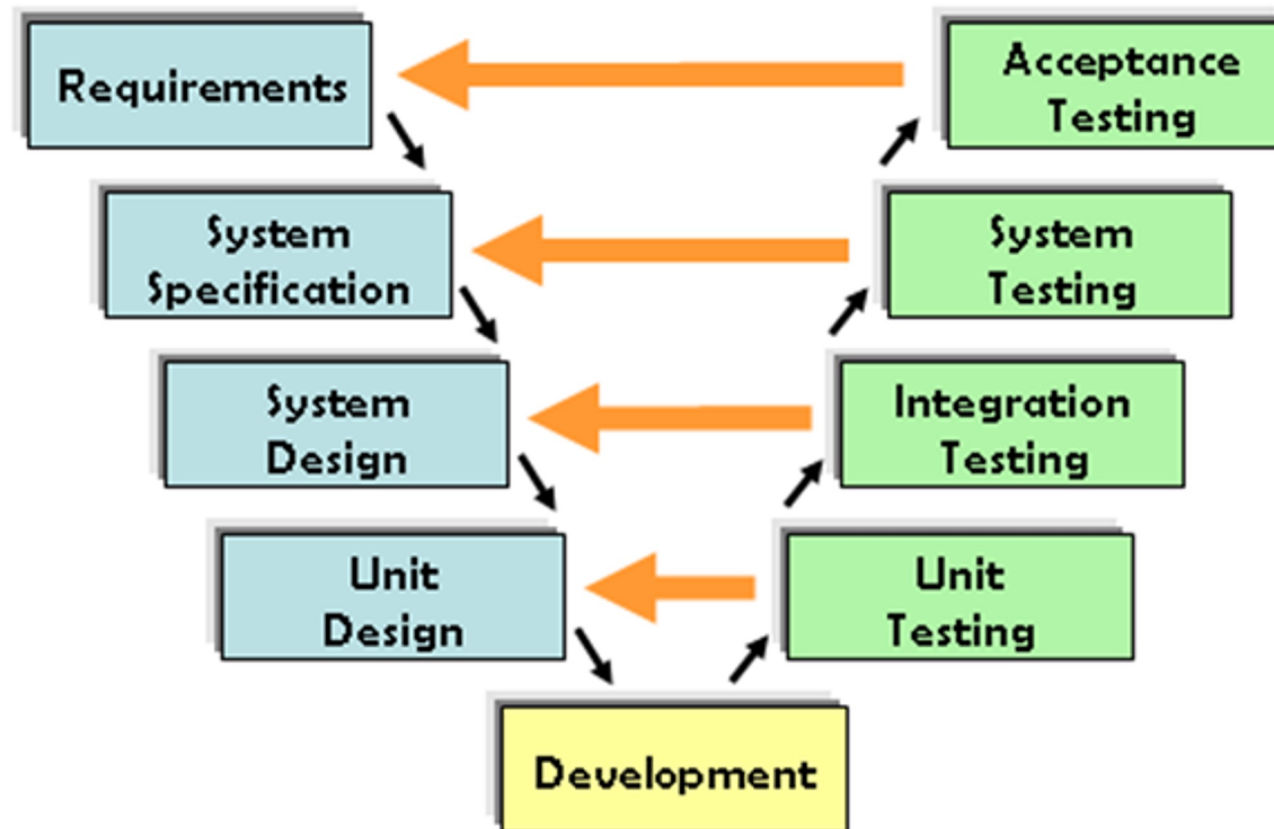
יתרונות

- פיתוח מסודר ולוגי
- תיעוד מלא
- מתאים לצוות לא מנוסה
- מתאים לפרויקטים צפויים ומובנים
- התהליך האידאלי אם אין שינויים באמצע

חסרונות

- הארכת זמן הפיתוח – דורש סבלנות מהלקוח
- חוסר גמישות לשינויים
- עלול לגרום לחוסר נצילות בצוות

V-model



יתרונות

- חסכון בכסף ומשאבים בשל תכנון זמן נכון - תחילת תהליך הבדיקות בשלב מוקדם של הפרויקט
- מבנה נכון של הבדיקות ויכולת כיסוי גבוהה בשל השלבים המקבילים המתבצעים בצד הפיתוח

חסרונות

- לא כולל את שלב התחזוקה



אז מה למדנו?

✓ מהם מודלים לפיתוח?

✓ מה ההבדל בין מודל מפל המים למודל V?

✓ כיצד נבדוק את המודלים?



מה נלמד במפגש הבא?

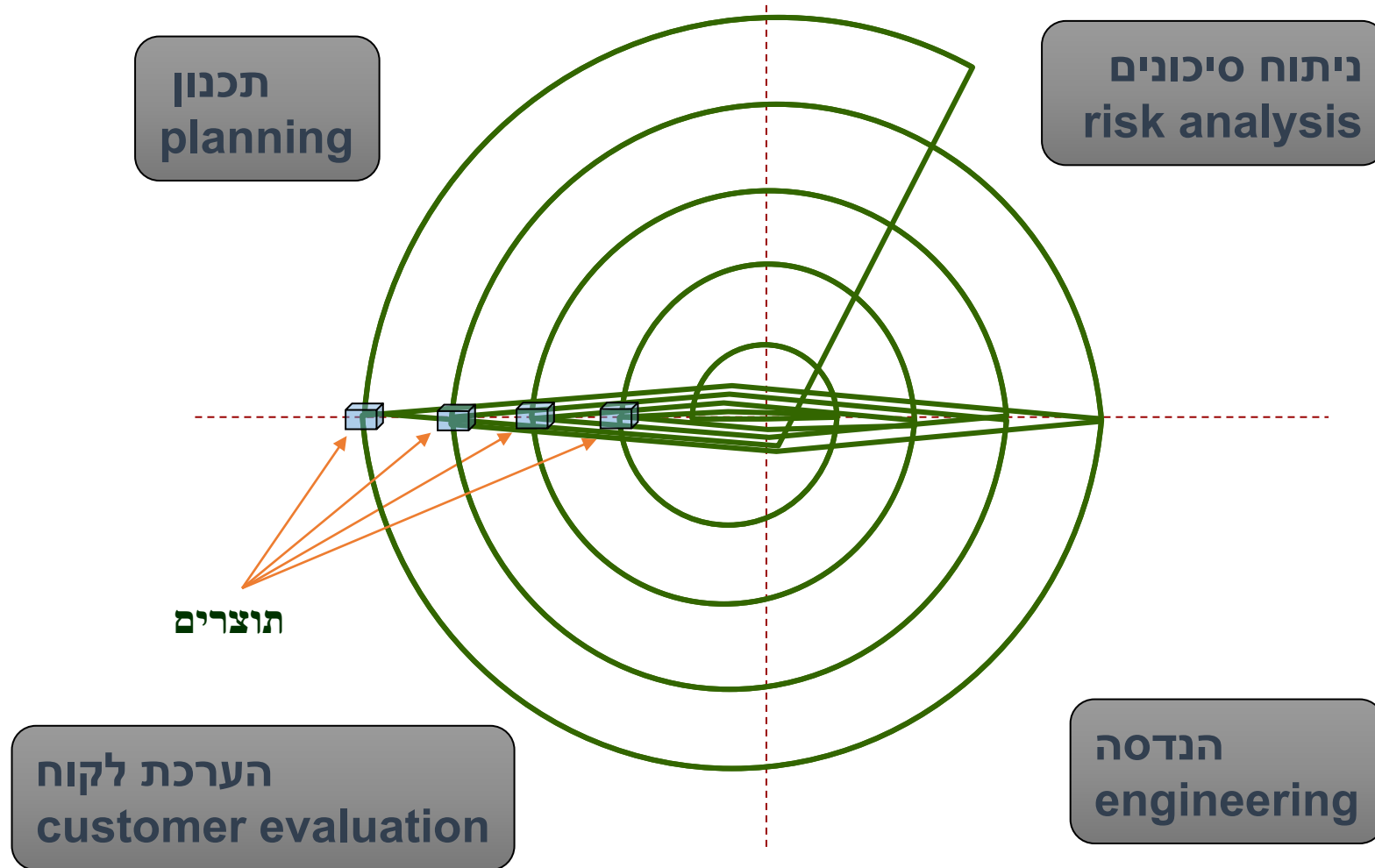
✓ המודל הספירלי

✓ המודל הזמיש

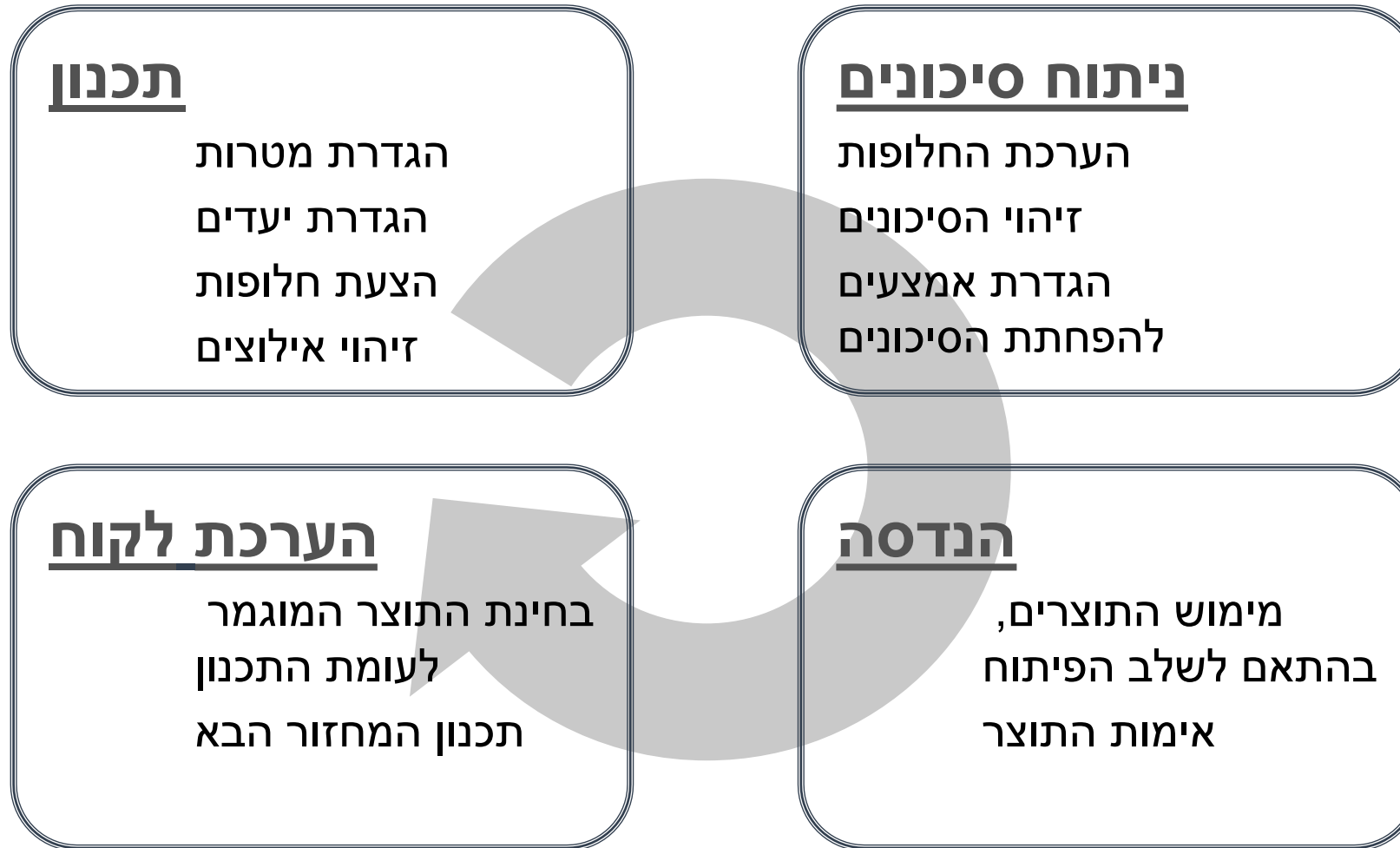
✓ סיכום המודלים



Spiral model



Spiral model - השלבים



Spiral model

יתרונות

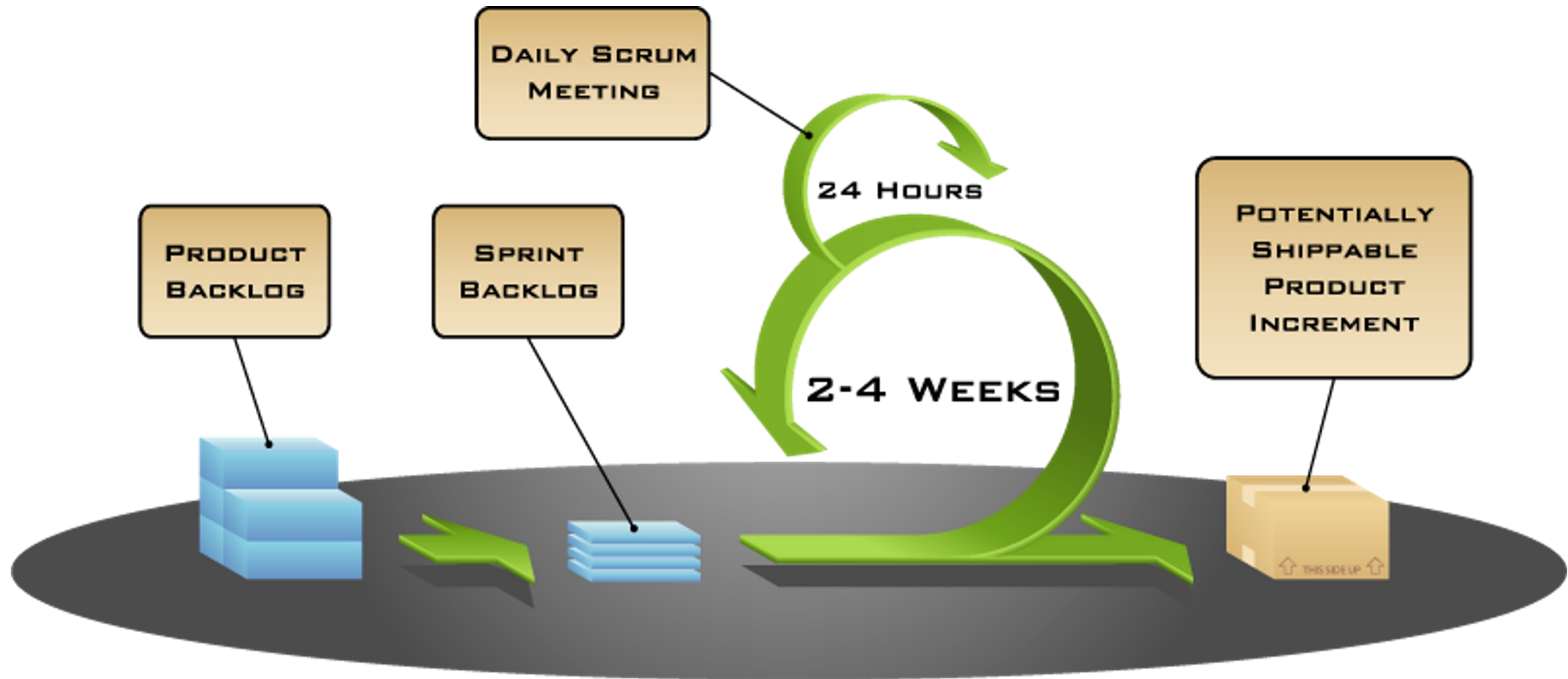
- מספק משוב וזיהוי מוקדם של בעיות
- מאפשר התאמה לשינויים
- מוריד סיכונים



חסרונות

- מצריך מומחיות בניהול סיכונים
- מייקר את התהליכים בשל הרחבת הניהול והתכנון

Agile model



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

עקרונות הפיתוח הזריז:

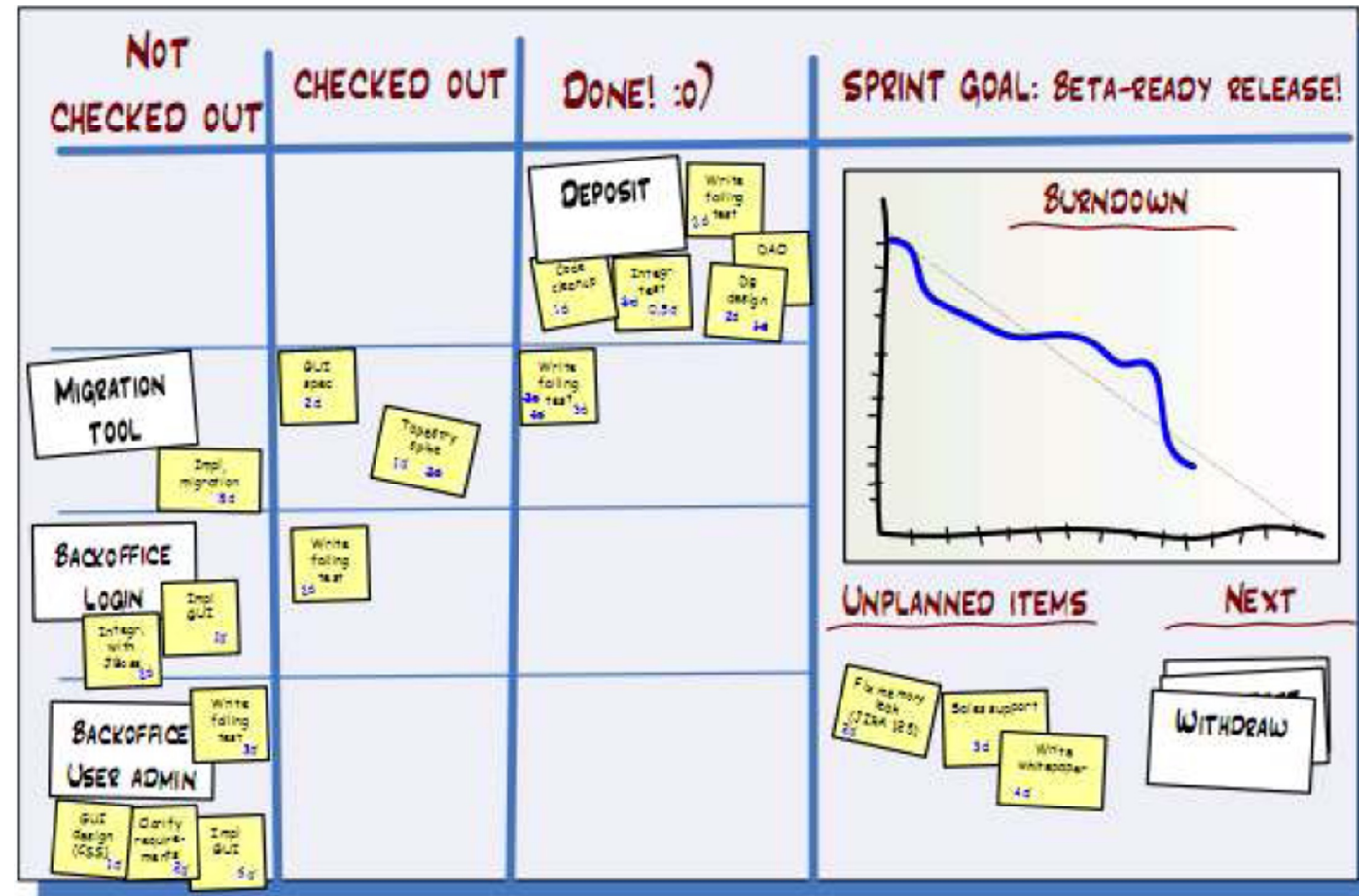
בני אדם ופעולות גומלין על פני תהליכים וכלים
תוכנה עובדת על פני תיעוד מקיף
שיתוף פעולה עם הלקוח על פני משא ומתן דרך חוזה
היענות לשינויים על פני התקדמות על פי התכנית

ניהול גרסאות, תהליכי עבודה, תהליכי עבודה, תהליכי עבודה

השוני מנקודת המבט של הבדיקות אל מול שאר המתודולוגיות:

- ריבוי גרסאות לבדיקה.
- תכולות בדיקה קטנות בכל "סבב".
- שינויים ותוספות רבות לאורך הפרויקט - מערכת בפיתוח דינמי.
- ניהול משימות וזמנים מתוכננים לטווחי זמן קצרים.
- יחס זמן תכנון מול ביצוע משתנה באופן מובהק.

Agile model



יתרונות

- דינמיות וגמישות
- איכות מוצר גבוהה – שילוב הבדיקות בשלב מוקדם, TDD
- הערכת זמנים מדויקת יותר – פיתוח בשלבים קטנים ותחומים
- תקשורת פעילה ורציפה בין כל הגורמים

חסרונות

- מתאים לפרויקטים קטנים עד בינוניים
- נדרש כ"א עצמאי ובעל ניסיון
- קושי בהסתכלות רחבה

אז מה למדנו?



✓ מהו המודל הספירלי?

✓ מהו המודל האגילי?

✓ במה שונה מודל ה-Agile משאר המודלים?

✓ אילו מודלים נוספים למדנו במפגש הקודם?

מה נלמד במפגש הבא?



✓ מהו מסמך דרישות?

✓ מהו מסמך האפיון?

✓ מהלך חיי פרויקט

✓ מהו מחיר התקלות?

✓ סוגי מערכות

QA?

THANK YOU